



TUGAS AKHIR - TE 145561

MONITORING PENGAMAN BEBAN LEBIH PANEL KONTROL Gardu Induk

Muhammad Ilham Fadlila
NRP 10311500010011

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 145561

***OVERLOAD MONITORING CONTROL PANEL IN
SUBSTATION***

Muhammad Ilham Fadlila
NRP 10311500010011

Supervisor
Ir. Josaphat Pramudijanto, M.Eng.

***ELECTRICAL AND AUTOMATION ENGINEERING DEPARTMENT
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2018***

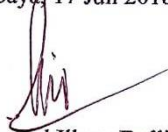
PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Monitoring Pengaman Beban Lebih Panel Kontrol Gardu Induk**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 17 Juli 2018



Muhammad Ilham Fadlila
NRP 10311500010011

-----Halaman ini sengaja dikosongkan-----

**MONITORING PENGAMAN BEBAN LEBIH PANEL KONTROL
GARDU INDUK**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya Teknik**

Pada

**Program Studi Elektro Industri
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Menyetujui :

Dosen Pembimbing


Ir. Josaphat Pramudijanto, M.Eng.
NIP. 19621005 199003 1 003

**SURABAYA
JULI, 2018**

-----Halaman ini sengaja dikosongkan-----

MONITORING PENGAMAN BEBAN LEBIH PANEL KONTROL GARDU INDUK

Nama Mahasiswa : Muhammad Ilham Fadlila
NRP : 10311500010011
Pembimbing : Ir. Josaphat Pramudijanto, M.Eng.
NIP : 19621005 199003 1 003

ABSTRAK

Dalam sistem pemeliharaan memantau arus dan tegangan pada pengaman beban lebih panel kontrol gardu induk belum tersedia. Untuk itu diperlukannya alat monitoring dalam proses pemantauan arus dan tegangan.

Proses monitoring tegangan dan arus diawali dengan pertama-tama sumber melewati MCB 2A sebagai pengaman, setelah itu melewati sensor arus ACS712 dan sensor tegangan ZMPT101B. Kemudian sumber 3 fasa ini dibagi menjadi 3 penyulang yang men-*supply* beban masing-masing 6x100 Watt per fasanya. Pada bagian penyulang juga dipasang PMT untuk keperluan proteksi yang diprogram oleh *Arduino* dan dapat dikontrol secara manual melalui panel kontrol. Kemudian pengolahan data dari sensor oleh *Arduino* dan kemudian ditampilkan ke LCD.

Alat pada Tugas Akhir ini dapat memonitoring pengaman beban lebih panel kontrol gardu induk. Ketika beban melebihi batas maksimal yang telah ditentukan yaitu sebesar 2A, maka panel pengaman akan memutus salah satu penyulang. Ketelitian pembacaan sensor arus dan tegangan sudah mendekati nilai yang terukur, dengan nilai persen eror pada masing-masing fasa di bawah 0,1%.

Kata Kunci: Monitoring, Gardu Induk, Sensor Arus, Sensor Tegangan.

-----Halaman ini sengaja dikosongkan-----

OVERLOAD MONITORING CONTROL PANEL IN SUBSTATION

Name : Muhammad Ilham Fadlila
Registration Number : 10311500010011
Supervisor : Ir. Josaphat Pramudijanto, M.Eng.
ID : 19621005 199003 1 003

ABSTRACT

In the maintenance system monitor the current and voltage on the load safety over the substation control panel is not yet available. Therefore, monitoring tools in monitoring the current and voltage are needed.

The process of monitoring the voltage and current begins with the first source passes MCB 2A as a safety, then passes through the ACS712 current sensor and ZMPT101B voltage sensor. Then the 3 phase source is divided into 3 feeders that supply each load of 6x100 Watt each phase. In the repeater section is also installed PMT for protective purposes programmed by Arduino and can be controlled manually through the control panel. Then processing data from the sensor by Arduino and then displayed to the LCD.

Results that can be obtained from overload monitoring control panel in substation. When the load exceeds the specified maximum limit that is equal to 2A, the safety panel will disconnect one of the feeders. The accuracy of the current and voltage sensor readings is close to the measured value, with a percent error value at each phase below 0.1%.

Keywords : Monitoring, Substation, Current Sensors, Voltage Sensors.

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Elektro Industri, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

MONITORING PENGAMAN BEBAN LEBIH PANEL KONTROL GARDU INDUK

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Josaphat Pramudijanto, M.Eng. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 17 Juli 2018

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN.....	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Sistematika Laporan.....	2
1.6 Relevansi.....	3
BAB II TEORI DASAR.....	5
2.1 Gardu Induk	5
2.1.1 Peralatan Utama GI.....	6
2.2 Proteksi Beban Lebih	8
2.2.1 Rele Arus Lebih (OCR)	9
2.3 Sensor Arus.....	11
2.3.1 <i>Current Transformer</i> (CT).....	11
2.4 Sensor Tegangan.....	13
2.5 <i>Arduino Mega 2560</i>	14
2.6 <i>Arduino IDE</i>	15
2.7 LCD 16x4	16
2.7.1 Cara Kerja LCD Secara Umum.....	17
BAB III PERANCANGAN ALAT	19
3.1 Diagram Fungsional Alat	19
3.2 Perancangan Perangkat Keras	20
3.2.1 Tata Letak (<i>Layout</i>)	20

3.2.2	<i>Wiring</i> Sistem AC.....	21
3.2.3	<i>Wiring</i> Sensor	22
3.2.4	<i>Mapping</i> Pin Amphenol	23
3.2.5	Modul Sensor Arus ACS712	25
3.2.6	Modul Sensor Tegangan ZMPT101B.....	27
3.3	Perancangan Perangkat Lunak.....	28
3.3.1	Program pada <i>Arduino</i>	28
3.3.2	Pemrograman pada Sensor Arus.....	33
3.3.3	Pemrograman pada Sensor Tegangan.....	34
BAB IV PENGUJIAN DAN ANALISA DATA.....		35
4.1	Pengujian <i>Hardware</i>	35
4.1.1	Pengujian PMT dan Panel Kontrol <i>Open-Close</i>	36
4.1.2	Pengujian Sensor Arus ACS712.....	38
4.1.3	Pengujian Sensor Tegangan ZMPT101B	41
4.1.4	Pengujian Tampilan LCD.....	43
4.1.5	Pengujian <i>Switch Selector</i>	44
4.2	Pengujian Keseluruhan	45
BAB V PENUTUP		47
5.1	Kesimpulan.....	47
5.2	Saran	47
DAFTAR PUSTAKA.....		49
LAMPIRAN A DATA		A-1
A.1	Data Sensor Arus	A-1
A.2	Data Sensor Tegangan	A-1
LAMPIRAN B PROGRAM.....		B-1
B.1	Program <i>Arduino</i>	B-1
LAMPIRAN C DATASHEET.....		C-1
C.1	<i>Datasheet</i> Sensor ACS712	C-1
C.2	<i>Datasheet</i> <i>Arduino Mega 2560</i>	C-7
RIWAYAT HIDUP PENULIS		D-1

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 Gardu Induk	5
Gambar 2.2 Transformator Tenaga.....	6
Gambar 2.3 PMT 20 kV	7
Gambar 2.4 PMS	7
Gambar 2.5 Panel Meter & Rele GI	8
Gambar 2.6 Karakteristik Rele <i>Instant</i>	9
Gambar 2.7 Karakteristik Rele <i>Definit Time</i>	9
Gambar 2.8 Karakteristik Rele <i>Inverse Time</i>	10
Gambar 2.9 Karakteristik Rele <i>Inverse Time</i>	10
Gambar 2.6 Karakteristik Rele Kombinasi <i>Inverse & Instant</i>	11
Gambar 2.11 Prinsip Kerja CT	12
Gambar 2.12 Proses Mengubah Arus AC Menjadi DC pada Sensor Arus.....	13
Gambar 2.13 Bentuk Tegangan DC dan AC	13
Gambar 2.14 Rangkaian <i>Peak Detector</i>	14
Gambar 2.15 <i>Software Arduino IDE</i>	16
Gambar 2.16 Skematik LCD 16x4	18
Gambar 3.1 Diagram Fungsional Alat.....	19
Gambar 3.2 Tata Letak Alat	20
Gambar 3.3 <i>Wiring</i> Sistem AC.....	21
Gambar 3.4 <i>Wiring</i> Sensor	22
Gambar 3.5 <i>Mapping</i> Pin Amphenol.....	23
Gambar 3.6 <i>Pin Out Diagram</i> Sensor ACS712.....	25
Gambar 3.7 Skematik Modul ACS712-5A.....	27
Gambar 3.8 Skematik Modul ZMPT101B	28
Gambar 3.9 <i>Flowchart</i> Kerja Alat	29
Gambar 3.10 Inisialisasi Program.....	30
Gambar 3.11 Program <i>Arduino Voice Setup</i>	32
Gambar 3.12 <i>Flowchart</i> Pemrograman Sensor Arus	33
Gambar 3.13 <i>Flowchart</i> Pemrograman Sensor Tegangan	34
Gambar 4.1 <i>Hardware</i> Keseluruhan	35
Gambar 4.2 Pengujian PMT dan Panel Kontrol <i>Open-Close</i>	36

Gambar 4.3	Program Pengujian PMT dan Panel Kontrol <i>Open-Close</i>	37
Gambar 4.4	Pengukuran Arus <i>Input</i> Sensor ACS712	39
Gambar 4.5	Program Pembacaan <i>Serial Monitor</i> Sensor Arus	39
Gambar 4.6	Hasil Pengujian Sensor Arus pada Fasa R	40
Gambar 4.7	Hasil Pengujian Sensor Arus pada Fasa S	40
Gambar 4.8	Hasil Pengujian Sensor Arus pada Fasa T	41
Gambar 4.9	Pengukuran Tegangan <i>Output</i> Sensor ZMPT101B	42
Gambar 4.10	Program Pembacaan <i>Serial Monitor</i> Sensor Tegangan ..	42
Gambar 4.11	Hasil Pengujian Sensor Tegangan	43
Gambar 4.12	Tampilan pada LCD	43
Gambar 4.13	Pengujian <i>Switch Selector</i>	44
Gambar 4.14	Hasil Pengujian Mode <i>Trip Fix</i>	45
Gambar 4.15	Hasil Pengujian Mode <i>Trip Rolling</i>	45

DAFTAR TABEL

	HALAMAN
Tabel 2.2	Spesifikasi <i>Arduino Mega 2560</i> 14
Tabel 2.3	Spesifikasi Kaki LCD 16x4 16
Tabel 3.1	<i>Data Sheet</i> Sensor ACS712 26
Tabel 4.1	Hasil Pengujian PMT dan Panel Kontrol <i>Open-Close</i> 38

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Semakin berkembangnya masyarakat terutama di kota besar menyebabkan bertambah pula kebutuhan akan energi listrik. Banyak dibangun industri-industri dan pusat perbelanjaan menyebabkan pertumbuhan beban listrik terus bergerak naik. PT. PLN (Perusahaan Listrik Negara) pun dituntut untuk bisa memenuhi tantangan tersebut. Kontinuitas pelayanan listrik pun menjadi hal yang sangat diutamakan. Dalam lingkup yang lebih kecil yaitu gardu induk (GI) dituntut untuk dapat beroperasi dengan handal. Namun dalam operasinya GI tidak lepas dari gangguan-gangguan yang bisa saja terjadi di lapangan. Salah satu gangguan yang sering terjadi antara lain gangguan arus lebih, karena kelebihan beban maupun gangguan hubung singkat pada penyulang ataupun busbar. Gangguan ini dapat menyebabkan kerusakan peralatan, berkurangnya *lifetime* peralatan, pemadaman secara individual maupun secara kolektif pada penyulang. Bahkan dapat menyebabkan pemadaman total (*black out*) pada GI jika pemutus tenaga (PMT) sisi *incoming*-nya lepas (*trip*).

Oleh karena itu dibuatlah tugas akhir yang mengacu pada judul tugas akhir “Rancang Bangun Simulasi Pengaman Beban Lebih Transformator Gardu Induk Menggunakan *Programmable Logic Controller*” milik Doni Irifan NRP 2210038020 [1]. Dari alat tersebut dikembangkan metode pengoperasiannya yang awalnya menggunakan PLC diubah menggunakan *Arduino* dan hasilnya dapat ditampilkan di LCD. Untuk menghindari terjadinya *black out* pada GI karena kelebihan beban, maka harus dilakukan pelepasan beban pada sisi penyulang. Dalam hal ini biasanya dipilih penyulang yang dianggap kurang penting. Artinya dalam proses pelepasan beban ini terdapat prioritas terhadap masing-masing penyulang. Untuk penyulang dengan pelanggan misal industri atau rumah sakit biasanya memiliki prioritas utama dan tidak boleh padam. Untuk itu penyulang yang dilepas biasanya penyulang dengan pelanggan rumah tangga. Pemadaman pun dilakukan secara bertahap dipadamkan tetapi masih terjadi kelebihan beban (*overload*) maka dapat dilakukan pemadaman pada penyulang dua, dan seterusnya.

Proses pelepasan beban seperti di atas sebenarnya dapat dilakukan secara manual oleh operator GI. Namun panel kontrol tersebut perlu dilakukannya *maintenance*. Sehingga dengan adanya permasalahan tersebut

perlu dibuat alat ini, namun permasalahan yang muncul adalah bagaimana cara untuk merancang alat monitor untuk pengukuran arus dan tegangan pada pengaman beban lebih panel kontrol gardu induk saat ada perawatan (*maintenance*), selanjutnya merancang komunikasi dari alat ke LCD, agar dapat memonitor arus dan tegangan pada pengaman beban lebih panel kontrol gardu induk saat dilakukannya perawatan dengan kontrol manual.

1.2 Permasalahan

Dalam sistem pemeliharaan memantau arus dan tegangan pada pengaman beban lebih panel kontrol gardu induk belum tersedia. Untuk itu diperlukannya alat monitoring dalam proses pemantauan arus dan tegangan. Proses pemantauan arus dan tegangan yang ada di dalam panel kontrol akan ditampilkan ke LCD.

1.3 Batasan Masalah

Pada tugas akhir ini, pengolahan data yang digunakan adalah *Arduino*, dengan memiliki batasan-batasan masalah sebagai berikut:

1. Panel kontrol gardu induk yang digunakan yaitu hanya sebagai simulator.
2. Tegangan yang digunakan 220 V.
3. Pengujian menggunakan beban lampu pijar 6x100 W tiap fasanya.

1.4 Tujuan

Tujuan yang akan dicapai dari Tugas Akhir ini adalah sebagai berikut:

1. Merancang alat monitoring pengaman beban lebih panel kontrol gardu induk.
2. Membuat alat monitoring pengaman beban lebih panel kontrol gardu induk.
3. Mengimplementasikan alat monitoring pengaman beban lebih panel kontrol gardu induk dalam masyarakat melalui PT. PLN (Perusahaan Listrik Negara) agar berguna untuk pemantuan arus dan tegangan saat terjadi gangguan maupun perawatan.

1.5 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka dari gardu induk, telemetering, *Arduino Mega 2560*, Sensor Arus ACS712, Sensor Tegangan ZMPT101B *software Arduino IDE*, Tampilan LCD.

Bab III Perancangan Sistem

Bab ini membahas perencanaan dan pembuatan perangkat keras (*Hardware*) yang meliputi desain alat serta pengimplementasian sensor dan aktuator yang digunakan, pengaturan modul Ethernet, dan pembuatan perangkat lunak (*Software*) yang meliputi program pada *Arduino IDE* untuk menjalankan alat tersebut, serta pembuatan program untuk tampilan di LCD.

Bab IV Pengujian dan Analisis

Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian sensor arus, sensor tegangan, serta pengujian keseluruhan alat terhadap beban lebih.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.6 Relevansi

Diharapkan dengan adanya alat monitoring pengaman beban lebih panel kontrol gardu induk dapat digunakan sebagai bahan pembelajaran mengenai penyulang.

-----Halaman ini sengaja dikosongkan-----

BAB II

TEORI DASAR

Pada bab ini membahas mengenai teori-teori dari peralatan yang digunakan dalam Tugas Akhir yang berjudul Telemetering Pengukuran Tegangan dan Arus Pada Panel Kontrol Beban Lebih dengan Komunikasi Wifi. Teori yang mendukung penyelesaian Tugas Akhir ini diantaranya adalah mengenai Gardu Induk, Proteksi Beban Lebih Gardu Induk, Sensor Arus, Sensor Tegangan, *Arduino*, dan LCD.

2.1 Gardu Induk [2]

Gardu Induk (GI) merupakan simpul didalam sistem tenaga listrik, yang terdiri dari susunan dan rangkaian sejumlah perlengkapan yang dipasang menempati suatu lokasi tertentu untuk menerima dan menyalurkan tenaga listrik, menaikkan dan menurunkan tegangan sesuai dengan tingkat tegangan kerjanya, tempat melakukan kerja *switching* rangkaian suatu sistem tenaga listrik dan untuk menunjang keandalan sistem tenaga listrik terkait. Gardu Induk dapat dilihat Pada Gambar 2.1.

Dalam sistem kelistrikan di PT. PLN (Persero) khususnya sistem Jawa – Bali terdapat beberapa jenis gardu induk sesuai dengan fungsi nya masing – masing yaitu :

- a. Gardu induk *Step-Up* berfungsi untuk menaikkan tegangan dari keluaran pembangkit (13,8 kV) ke sistem jaringan interkoneksi 500 kV.
- b. Gardu induk TET (Tegangan Ekstra Tinggi) 500/150 kV, GI ini berfungsi untuk menurunkan tegangan (*Step-Down*) dari sistem interkoneksi ke jaringan tegangan yang lebih rendah yaitu jaringan 150 kV.



Gambar 2.1 Gardu Induk.

- c. Gardu induk TT (Tegangan Tinggi) 150/20 kV, gardu induk jenis ini berfungsi untuk mengkonversi tegangan dari jaringan 150 kV ke jaringan saluran udara tegangan menengah (SUTM) 20 kV.

2.1.1 Peralatan Utama GI

Pada suatu gardu induk terdapat peralatan – peralatan listrik yang menunjang kinerja GI tersebut. Peralatan GI yang utama antara lain :

A. Transformator Tenaga

Transformator (Trafo) tenaga adalah suatu peralatan tenaga listrik yang berfungsi untuk mentransformasikan daya listrik dari tegangan tinggi ke tegangan rendah atau sebaliknya.

Dalam operasi penyaluran tenaga listrik transformator dapat dikatakan sebagai jantung dari transmisi dan distribusi. Dalam kondisi ini suatu transformator diharapkan dapat beroperasi secara maksimal. Untuk bentuk fisik dari Transformator Tenaga dapat dilihat Pada Gambar 2.2.

B. Pemutus Tenaga (PMT) & Pemisah (PMS)

Pemutus Tenaga (PMT) atau CB (*Circuit Breaker*) adalah salah satu peralatan yang terpasang di Gardu Induk, berfungsi untuk menghubungkan atau memutuskan arus beban dan arus gangguan sesuai dengan *rating*nya.



Gambar 2.2 Transformator Tenaga.

Kecepatan untuk melepas maupun menghubungkan beban dengan sumber tergantung dari kecepatan PMT tersebut. Waktu antara *tripping coil* mulai di *energized* sampai kontak PMT bergerak disebut waktu buka (*Opening time*). Waktu antara kontak PMT bergerak sampai busur api padam disebut waktu busur api (*Arcing time*). Jumlah waktu dari *tripping coil* mulai di *energized* sampai sampai busur api padam disebut total waktu buka (*Total break time*). Untuk bentuk fisik dari PMT dapat dilihat Pada Gambar 2.3.

Pemisah (PMS) merupakan peralatan yang dirancang untuk memutus atau menyambung sirkit listrik tanpa arus atau dengan arus yang relatif kecil dibandingkan dengan arus nominal beban. PMS merupakan saklar pemisah yang digunakan untuk membebaskan bagian-bagian suatu instalasi dari tegangan, atau untuk menukar hubungan.



Gambar 2.3 PMT 20 kV.



Gambar 2.4 PMS.



Gambar 2.5 Panel Meter & Rele GI.

Saklar-saklar ini memiliki kontak-kontak berpegas. Pisau-pisau saklar dijepit antara kontak-kontak itu. Konstruksi kontak-kontak ini harus sedemikian hingga tidak mungkin membuka karena pengaruh gaya-gaya yang timbul kalau terjadi hubungan singkat. Penggerak PMS dapat berupa mekanik yang dilakukan secara manual, atau energi listrik dengan motor-motor listrik ataupun dengan *pneumatic*. Untuk bentuk fisik dari PMS dapat dilihat Pada Gambar 2.4.

C. Sistem Metering & Proteksi

Sistem metering pada tegangan menengah (TM) maupun tegangan tinggi (TT) seperti pada GI dilakukan dengan bantuan transformator *instrument* berupa *Current Transformer* (CT) untuk mengukur nilai arus dan *Potential Transformer* (PT) untuk mengukur nilai tegangan.

Nilai pembacaan arus dan tegangan ini selain untuk keperluan *monitoring* daya dan beban, juga digunakan untuk keperluan proteksi dengan bantuan rele. Baik proteksi terhadap gangguan arus maupun gangguan tegangan.

Selain itu pada gardu induk, peralatannya juga diproteksi dari gangguan yang lain seperti panas berlebih, *under & over* frekuensi, gangguan tekanan pada tangki transformator dll. Pada Gambar 2.5 adalah gambar dari panel meter.

2.2 Proteksi Beban Lebih GI [3]

Agar penyaluran energi listrik tetap terjamin kontinuitasnya serta aman terhadap lingkungan dan peralatan maka diperlukan peralatan yang

dapat mengamankan/memproteksi sistem GI. Peralatan yang dimaksud adalah rele proteksi. Salah satu gangguan yang sering terjadi di GI adalah gangguan arus lebih. Gangguan ini bisa disebabkan oleh gangguan beban lebih (*Overload*) atau hubung singkat baik antar fasa maupun fasa ke tanah.

2.2.1 Rele Arus Lebih (OCR)

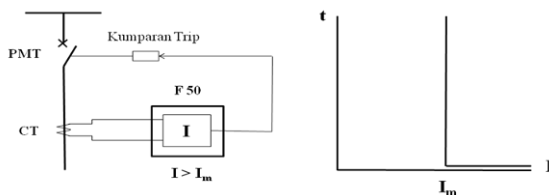
Rele Arus Lebih / *Over Current relay* merupakan rele proteksi yang bekerja berdasarkan parameter arus yang dibaca melalui sensor arus berupa CT (*Current Transformer*). Apabila Arus beban melebihi *setting* yang telah ditentukan maka rele akan mengirimkan sinyal *trip* ke PMT.

Berdasarkan *Standard IEC 60225* karakteristik kerja arus – waktu, rele arus lebih dibagi atas beberapa jenis, yaitu :

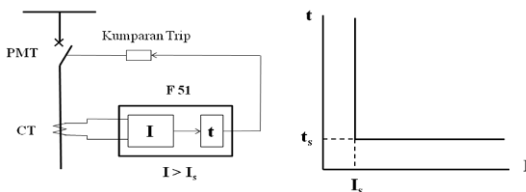
A. Rele Arus Lebih Seketika (*Moment, Instant*)

Karakteristik rele ini bekerja tanpa adanya penundaan waktu. Jangka waktu mulai rele *pick up* sampai kerja rele sangat singkat (20 ms – 50 ms). Dapat dilihat Pada Gambar 2.6.

B. Rele Arus Lebih Waktu tertentu (*Definit Time*)



Gambar 2.6 Karakteristik Rele *Instant*.



Gambar 2.7 Karakteristik Rele *Definit Time*.

Karakteristik rele ini bekerja dengan settingan waktu. Jangka waktu mulai rele *pick up* sampai kerja rele diperpanjang dengan harga tertentu tidak tergantung besarnya arus. Dapat dilihat Pada Gambar 2.7.

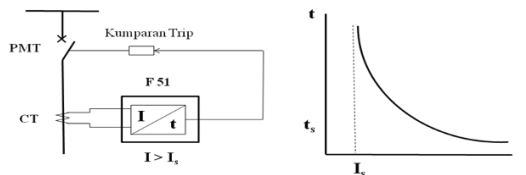
C. Rele Arus Lebih Waktu Terbalik (*Inverse Time*)

Karakteristik rele ini bekerja berkebalikan dengan waktu. Jangka waktu mulai rele *pick up* sampai kerja rele berbanding terbalik dengan besarnya arus yang mengerjakannya.

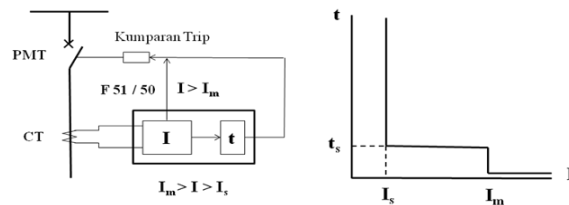
Karakteristik rele *inverse* dibedakan menjadi 3 jenis yaitu *Standard Inverse* (SI), *Very Inverse* (VI) dan *Extremely inverse* (EI). Dapat dilihat Pada Gambar 2.8.

D. Kombinasi Rele Arus Lebih Seketika dengan Waktu Tertentu

Rele ini bekerja dengan perpaduan 2 karakteristik. Apabila arus yang melewati melebihi $I_{setting}$ rele *instant* dan *definit*, maka kedua karakteristik tersebut sama – sama bekerja tetapi rele *instant* lebih cepat bekerja maka indikasi rele menunjukkan *instant* sedangkan apabila arus yang melewati melebihi $I_{setting}$ rele *definit* tetapi dibawah $I_{setting}$ *instant* maka yang merasakan dan bekerja adalah karakteristik rele *definit* sedangkan karakteristik rele *instant* tidak bekerja. Dapat dilihat Pada Gambar 2.9.



Gambar 2.8 Karakteristik Rele *Inverse Time*.



Gambar 2.9 Karakteristik Rele Kombinasi *Instant* & DT.

E. Kombinasi Rele Arus Lebih Seketika dengan Waktu Terbalik

Rele ini bekerja dengan perpaduan 2 karakteristik. Apabila arus yang melewati melebihi $I_{setting}$ rele *instant* dan *inverse*, maka kedua karakteristik tersebut sama – sama bekerja tetapi rele *instant* lebih cepat bekerja maka indikasi rele menunjukkan *instant* sedangkan apabila arus yang melewati melebihi $I_{setting}$ rele *inverse* tetapi dibawah $I_{setting}$ *instant* maka yang merasakan dan bekerja adalah karakteristik rele *inverse* sedangkan karakteristik rele *instant* tidak bekerja. Dapat dilihat Pada Gambar 2.10.

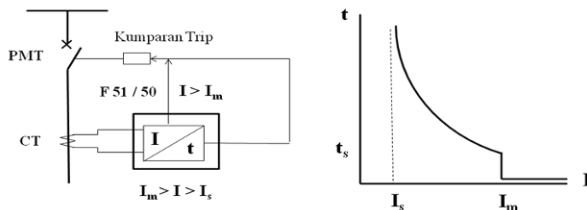
2.3 Sensor Arus [4]

Sensor arus merupakan suatu piranti yang digunakan untuk mengukur besaran arus pada suatu sistem listrik. Sensor ini mengubah arus menjadi suatu besaran listrik (arus / tegangan) yang lebih proporsional untuk selanjutnya digunakan untuk keperluan *metering* maupun proteksi. Contoh sensor arus adalah CT dan ACS712.

Perlu diketahui dalam sistem AC, hasil pembacaan arus pada alat ukur merupakan nilai RMS (*root means square*) yang merupakan nilai efektifnya. Besar nilai arus RMS adalah $\sqrt{2} \times$ Arus puncaknya ($I_{RMS} = \sqrt{2} \times I_{PEAK}$) atau sekitar 0,707 arus puncak.

2.3.1 Current Transformer (CT)

Trafo arus digunakan untuk pengukuran arus yang besarnya ratusan ampere lebih yang mengalir pada jaringan tegangan tinggi. Jika arus hendak diukur mengalir pada tegangan rendah dan besarnya dibawah 5 Ampere, maka pengukuran dapat dilakukan secara langsung sedangkan arus yang besar tadi harus dilakukan secara tidak langsung dengan menggunakan trafo arus sebutan trafo pengukuran arus yang besar.

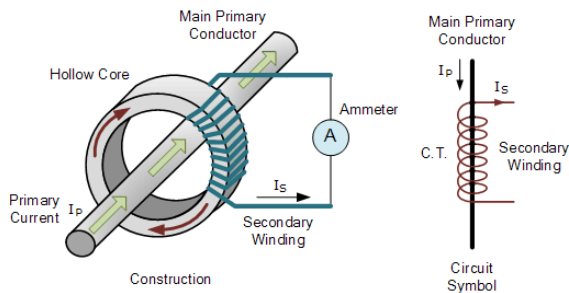


Gambar 2.10 Karakteristik Rele Kombinasi *Inverse & Instant*.

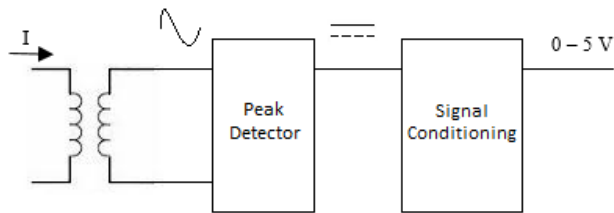
Prinsip kerja CT dapat dilihat Pada Gambar 2.11 sama dengan trafo daya satu fasa. Jika pada kumparan primer mengalir arus I_p , maka pada kumparan primer akan timbul gaya gerak magnet sebesar $N_p I_p$, gaya gerak magnet ini memproduksi fluks pada inti. Fluks ini membangkitkan gaya gerak listrik pada kumparan sekunder. Jika kumparan sekunder tertutup, maka pada kumparan sekunder mengalir arus I_s . arus ini menimbulkan gaya gerak magnet $N_s I_s$ pada kumparan sekunder.

Perbedaan utama trafo arus dengan trafo daya adalah jumlah belitan primer sangat sedikit, tidak lebih dari 5 belitan. Arus primer tidak mempengaruhi beban yang terhubung pada kumparan sekundernya, karena arus primer ditentukan oleh arus pada jaringan yang diukur. semua beban pada kumparan sekunder dihubungkan seri. terminal sekunder trafo tidak boleh terbuka, oleh karena itu terminal kumparan sekunder harus dihubungkan dengan beban atau dihubung singkat jika bebannya belum dihubungkan.

Sensor arus berupa *current transformer* mendapatkan input arus AC, kemudian agar dapat dibaca oleh mikrokontroler maka arus output harus diubah menjadi DC dengan menggunakan rangkaian *peak detector*. Agar output yang dihasilkan sebesar 0-5 A maka diperlukan rangkaian *signal conditioning*. Proses perubahan arus AC menjadi DC dapat dilihat Pada Gambar 2.12.



Gambar 2.11 Prinsip Kerja CT.

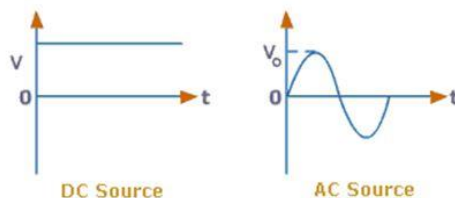


Gambar 2.12 Proses Mengubah Arus AC Menjadi DC pada Sensor Arus

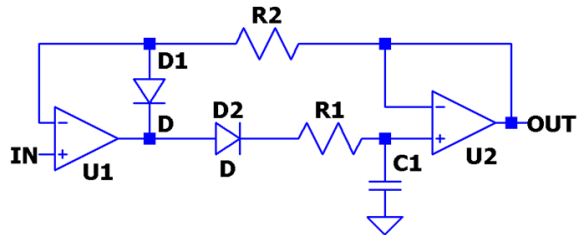
2.4 Sensor Tegangan [5]

Sensor tegangan merupakan piranti yang umum digunakan pada peralatan elektronik. Secara sederhana sensor tegangan bisa didapatkan melalui perancangan rangkaian pembagi tegangan, dan juga dapat didesain menggunakan sensor tegangan dengan menggunakan pembagi tegangan dapat digunakan pada tegangan AC maupun DC, sedangkan sensor yang menggunakan transformator hanya dapat digunakan untuk men-sensing tegangan AC.

Terlepas dari jenis sensor tegangan yang digunakan, dalam aplikasi berbasis mikroprosesor teknik pembacaan tegangan AC dan DC sangatlah berbeda. Tegangan DC seperti terlihat pada gambar 2 memiliki sifat nilai yang relatif konstan, selain itu tegangan DC pada umumnya hanya berada pada satu kuadran (positif saja atau negatif saja). Dengan sifat/karakteristik tegangan seperti di atas pembacaan tegangan DC menjadi mudah untuk diterapkan. Berbeda dengan tegangan AC, bentuk tegangannya tidak konstan melainkan mengikuti bentuk sinus. Selain itu tegangan AC berada pada dua kuadran, positif maupun negatif, sehingga tidak dapat langsung diberikan ke pin input mikroprosesor. Bentuk tegangan DC dan AC dapat dilihat Pada Gambar 2.13.



Gambar 2.13 Bentuk tegangan DC dan AC.



Gambar 2.14 Rangkaian *Peak Detector*.

Untuk menghasilkan tegangan output DC pada sensor tegangan diperlukannya rangkaian *peak detector*. Rangkaian *peak detector* adalah rangkaian yang terdiri dari dioda dan kapasitor yang dihubungkan seri yang menghasilkan output berupa tegangan DC yang sama dengan tegangan AC sebagai input. Rangkaian *peak detector* dapat dilihat Pada Gambar 2.14.

2.5 Arduino Mega 2560 [6]

Arduino merupakan perangkat elektronik atau papan rangkaian elektronik *open – source* yang di dalamnya terdapat komponen utama, yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel. Mikrokontroler itu sendiri adalah *chip* atau IC (*Integrated circuit*) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan. Jadi mikrokontroler bertugas sebagai otak yang mengendalikan *input*, proses, dan *output* sebuah rangkaian elektronik.

Tabel 2.1 Spesifikasi *Arduino Mega 2560*.

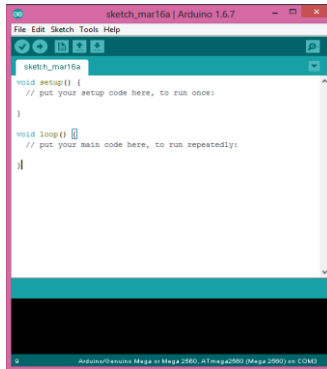
Spesifikasi	Keterangan
Chip Mikrokontroler	ATMega2560
Tegangan Operasi	5 V
Tegangan <i>Input</i> (Rekomendasi)	7 V-12 V
Tegangan <i>Input</i> (Limit)	6 V- 20 V
Pin Digital I/O	54, (15 buah diantaranya dapat digunakan sebagai <i>Output PWM</i>)
Pin Analog <i>Input</i>	16 (A0 – A.15)

Spesifikasi	Keterangan
Arus DC per Pin I/O	40 mA
Arus DC Pin 3,3V	50 mA
Memori <i>Flash</i>	256 KB, 8 KB telah digunakan untuk <i>Bootloader</i>
SRAM	8 Kb
EEPROM	4 Kb
<i>Clock Speed</i>	16 Hz

Tipe *Arduino* yang akan digunakan pada penelitian kali ini yaitu *Arduino Mega 2560*. *Arduino Mega 2560* adalah mikrokontroler berbasis ATmega2560 dengan *Clock Speed* 16Mhz dan *Flash Memory* 256KB. Tegangan operasi untuk *Arduino* jenis ini yaitu 5 V. Sedangkan tegangan *input* yang direkomendasikan yakni 7 – 12 V. *Arduino* ini memiliki 54 pin digital *input/output* pada pin 22-53 dengan 15 pin diantaranya merupakan pin PWM pada pin 0-13, 16 pin analog *input* pada pin A0 – A15, sambungan USB, sambungan catu daya tambahan dan tombol pengaturan ulang. Pada Tabel 2.1 merupakan spesifikasi dari *Arduino Mega 2560*.

2.6 *Arduino IDE* [7]

Arduino IDE adalah perangkat lunak yang digunakan untuk memprogram, monitoring dan *debugging* mikrokontroler *Arduino*. IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah *Arduino* dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. *Arduino* menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman *Arduino* (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler *Arduino* telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler Arduino* dengan mikrokontroler. Pada Gambar 2.15 merupakan tampilan awal untuk membuat program pada *software Arduino IDE*.



Gambar 2.15 Software Arduino IDE.

2.7 LCD 16x4 [8]

Display elektronik adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD (*Liquid Cristal Display*) adalah salah satu jenis display elektronik yang dibuat dengan teknologi CMOS logic yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap front-lit atau mentransmisikan cahaya dari back-lit. LCD (*Liquid Cristal Display*) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik. Untuk bentuk fisik dari LCD 16x4 dapat dilihat Pada Tabel 2.2 adalah spesifikasi kaki LCD 16x4.

Adapun fitur yang disajikan dalam LCD ini adalah :

- Terdiri dari 16 karakter dan 2 baris.
- Terdapat karakter generator terprogram.
- Dapat dialamati dengan mode 4-bit dan 8-bit.
- Dilengkapi dengan *backlight*.

Tabel 2.2 Spesifikasi Kaki LCD 16 x 4.

Pin	Deskripsi
1	Ground
2	Vcc
3	Pengatur kontras
4	“RS” Instruction/Register Select
5	“R/W” Read/Write LCD Registers

Pin	Deskripsi
6	“EN” Enable
7	Data I/O Pins
8	Data I/O Pins
9	Data I/O Pins
10	Data I/O Pins
11	Data I/O Pins
12	Data I/O Pins
13	Data I/O Pins
14	Data I/O Pins
15	Vcc
16	Ground

2.7.1 Cara Kerja LCD Secara Umum

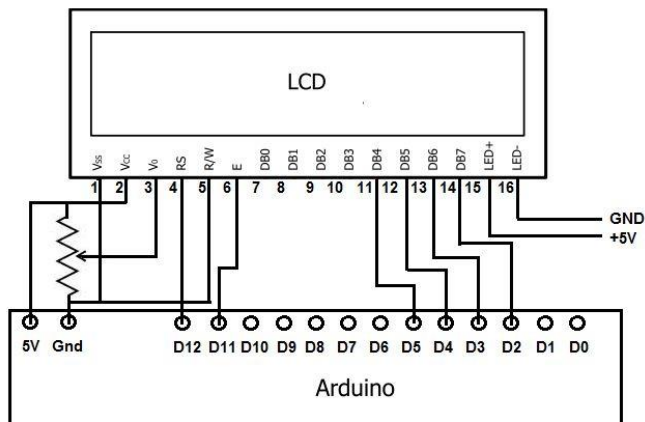
Pada aplikasi umumnya RW diberi logika rendah “0”. Bus data terdiri dari 4-bit atau 8-bit. Jika jalur data 4-bit maka yang digunakan ialah DB4 sampai dengan DB7. Sebagaimana terlihat pada tabel deskripsi, interface LCD merupakan sebuah paralel *bus*, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8-bit dikirim ke LCD secara 4-bit atau 8-bit pada satu waktu. Jika mode 4-bit yang digunakan, maka 2 nibble data dikirim untuk membuat sepenuhnya 8-bit (pertama dikirim 4-bit MSB lalu 4-bit LSB dengan pulsa *clock* EN setiap nibblenya). Jalur kontrol EN digunakan untuk memberitahu LCD bahwa mikrokontroler mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus mengatur EN ke kondisi high “1” dan kemudian mengatur dua jalur kontrol lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data *bus*.

Saat jalur lainnya sudah siap, EN harus diatur ke “0” dan tunggu beberapa saat (tergantung pada *datasheet* LCD), dan *setting* EN kembali ke *high* “1”. Ketika jalur RS berada dalam kondisi *low* “0”, data yang dikirimkan ke LCD dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan layar, posisi kursor dll). Ketika RS dalam kondisi *high* atau “1”, data yang dikirimkan adalah data ASCII yang akan ditampilkan di layar. Misal, untuk menampilkan huruf “A” pada layar maka RS harus diatur ke “1”. Jalur kontrol R/W harus berada dalam kondisi *low* (0) saat informasi pada data *bus* akan dituliskan ke LCD. Apabila R/W

berada dalam kondisi *high* “1”, maka program akan melakukan pembacaan data dari LCD. Instruksi pembacaan hanya satu, yaitu mendapatkan status LCD (membaca status LCD), lainnya merupakan instruksi penulisan. Jadi hampir setiap aplikasi yang menggunakan LCD, R/W selalu diatur ke “0”. Jalur data dapat terdiri 4 atau 8 jalur (tergantung mode yang dipilih pengguna), DB0, DB1, DB2, DB3, DB4, DB5, DB6 dan DB7. Mengirim data secara paralel baik 4-bit atau 8-bit merupakan 2 mode operasi primer. Untuk membuat sebuah aplikasi *interface* LCD, menentukan mode operasi merupakan hal yang paling penting.

Mode 8-bit sangat baik digunakan ketika kecepatan menjadi keutamaan dalam sebuah aplikasi dan setidaknya minimal tersedia 11 pin I/O (3 pin untuk kontrol, 8 pin untuk data). Sedangkan mode 4-bit minimal hanya membutuhkan 7-bit (3 pin untuk kontrol, 4 pin untuk data). Bit RS digunakan untuk memilih apakah data atau instruksi yang akan ditransfer antara mikrokontroller dan LCD. Jika bit ini diatur (RS = 1), maka bit pada posisi kursor LCD saat itu dapat dibaca atau ditulis. Jika bit ini diatur kembali (RS = 0), merupakan instruksi yang dikirim ke LCD atau status eksekusi dari instruksi terakhir yang dibaca.

Untuk gambar skematik LCD 16x4 dapat dilihat Pada Gambar 2.16.



Gambar 2.16 Skematik LCD 16 x 4.

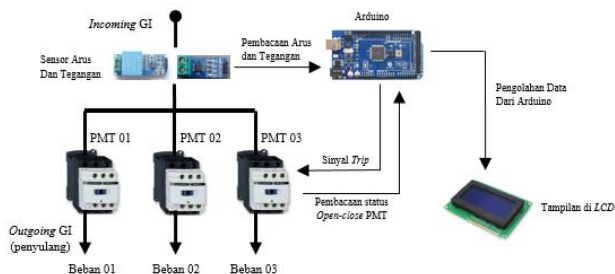
BAB III PERANCANGAN ALAT

Perancangan Alat Monitoring Pengaman Beban Lebih Panel Kontrol Gardu Induk Menggunakan *Arduino* ini diawali dengan diagram fungsional alat yang menunjukkan konsep dan cara kerja alat pada tugas akhir ini. Kemudian di jelaskan tentang perancangan perangkat keras (*hardware*) meliputi perancangan tata letak alat, perancangan modul sensor arus, modul Pemutus Tenaga (PMT), Panel Kontrol *Open – Close* PMT, & Perancangan I/O Peralatan ke *Arduino*. Dan yang terakhir adalah pemaparan rancangan perangkat lunak (*software*) yang terdiri dari perancangan program *Arduino*.

Proses monitoring tegangan dan arus diawali dengan pertama-tama sumber melewati MCB 2A sebagai pengaman, setelah itu melewati sensor arus ACS712 dan sensor tegangan ZMPT101B. Kemudian sumber 3 fasa ini dibagi menjadi 3 penyulang yang mensupply beban masing-masing 6x100 Watt per fasanya.

Pada bagian penyulang juga dipasang PMT untuk keperluan proteksi yang diprogram oleh *Arduino* dan dapat dikontrol secara manual melalui panel kontrol. Kemudian pengolahan data dari sensor oleh *Arduino* dan kemudian ditampilkan ke LCD. Apabila ada beban yang melebihi dari batas yang sudah ditentukan maka ada salah satu penyulang yang akan diputus dan status dari penyulang tersebut dapat diketahui pada tampilan LCD. Diagram fungsional alat dapat dilihat Pada Gambar 3.1.

3.1 Diagram Fungsional Alat



Gambar 3.1 Diagram Fungsional Alat.[9]

Dari diagram fungsional alat seperti Gambar 3.1 dapat dilihat bahwa cara kerja alat ini adalah sensor arus membaca nilai arus *incoming* yang kemudian hasil pembacaan data *analog* arus masuk ke pin AI pada *Arduino* untuk di ubah menjadi data *digital* yang kemudian diolah dan dapat ditampilkan di LCD.

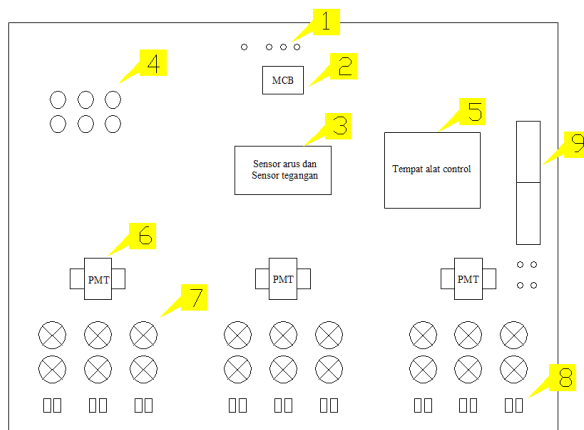
3.2 Perancangan Perangkat Keras

Perangkat keras yang di buat dalam merupakan suatu miniatur peralatan yang ada di gardu induk seperti sensor arus, PMT, panel kontrol *open – close* dll. Semua perangkat ini di integrasikan sehingga dapat menjalankan suatu mekanisme kerja seperti di gardu induk nyata.

3.2.1 Tata Letak (*Layout*)

Perancangan tata letak dimaksudkan agar penempatan peralatan menjadi rapi dan mudah dimengerti sehingga dari segi estetika dan fungsinya menjadi lebih baik. Pada Gambar 3.2 merupakan gambar tata letak alat. Dari gambar tersebut diperoleh keterangan seperti berikut :

1. Terminal *Input* tegangan 3 fasa (N-R-S-T).
2. *Mini Circuit Breaker* 2A, Berfungsi sebagai pengaman peralatan.
3. 3 Sensor Arus dan 3 Sensor Tegangan, berfungsi untuk mengukur nilai arus dan tegangan masing-masing fasa (R-S-T).

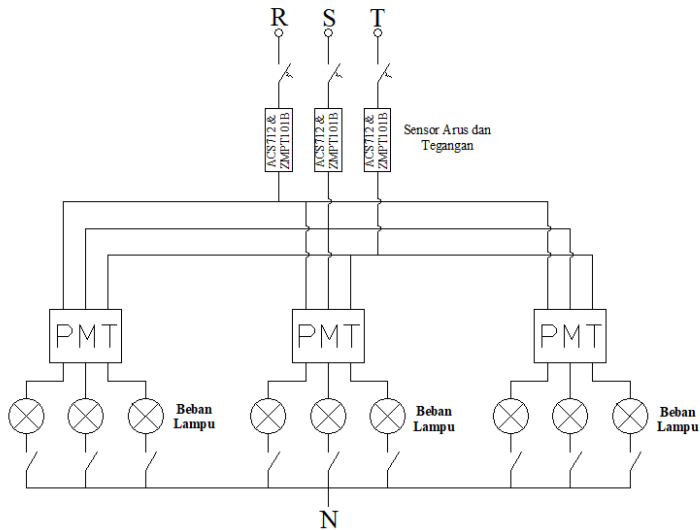


Gambar 3.2 Tata Letak Alat.

4. Panel Kontrol *Open-Close* PMT, untuk melepas dan memasukkan PMT secara manual.
5. Box tempat alat-alat kontrol (*Arduino*, *Ethernet*, dan rangkaian penunjang), Switch selektor, dan LCD
6. PMT, berfungsi sebagai pemutus beban.
7. Beban Lampu Pijar masing-masing 100W/220V.
8. Saklar lampu, berfungsi untuk mengatur nyala/mati lampu.
9. Terminal pin *output* sensor (AI), *Open-Close* PMT (DO & DI) serta terminal pin *supply* DC (24V & 5V).

3.2.2 Wiring Sistem AC

Jika dilihat Pada Gambar 3.3 Sumber 3 fasa (R-S-T) pertama-tama melewati MCB 2A sebagai pengaman, setelah itu melewati sensor arus ACS712 dan sensor tegangan ZMPT101B. Kemudian sumber 3 fasa ini dibagi menjadi 3 penyulang yang memberi *supply* beban masing-masing 2x100 Watt per fasanya.



Gambar 3.3 Wiring Sistem AC.

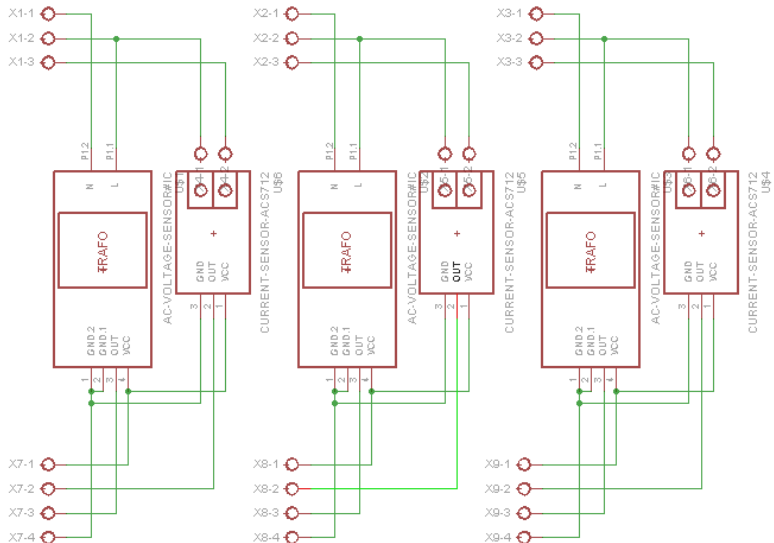
Pada bagian penyulang juga dipasang PMT untuk keperluan proteksi yang diprogram oleh *Arduino* dan dapat dikontrol secara manual melalui panel kontrol.

Wiring untuk beban pada alat ini menggunakan kabel NYAF 0.75 mm dihubungkan secara Y (bintang) di mana setelah melewati lampu kabel R-S-T dihubungkan ke titik Netral (N).

3.2.3 Wiring Sensor

Jika dilihat Pada Gambar 3.4 ada 3 bagian sensor yang tiap bagian terdapat sensor arus ACS712-5A dan sensor tegangan ZMPT101B. Tiap-tiap bagian tersebut terdapat di tiap fasa. Dari gambar tersebut diperoleh keterangan seperti berikut :

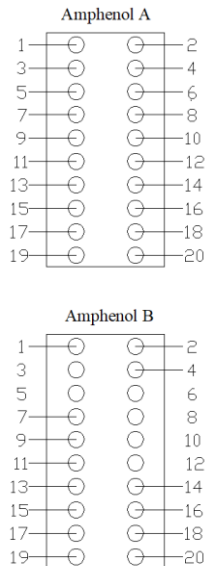
- X1-1 → Masukkan dari N
- X1-2 → Masukkan dari MCB 2A (R)
- X1-3 → Menuju ke beban
- X2-1 → Masukkan dari N



Gambar 3.4 Wiring Sensor.

X2-2 → Masukkan dari MCB 2A (S)
 X2-3 → Menuju ke beban
 X3-1 → Masukkan dari N
 X3-2 → Masukkan dari MCB 2A (T)
 X3-3 → Menuju ke beban
 X7-1 → Masuk ke pin *Arduino* GND
 X7-2 → *Output* data I (R) masuk ke pin *Arduino*
 X7-3 → *Output* data V (R) masuk ke pin *Arduino*
 X7-4 → Masuk ke pin *Arduino* VCC
 X8-1 → Masuk ke pin *Arduino* GND
 X8-2 → *Output* data I (S) masuk ke pin *Arduino*
 X8-3 → *Output* data V (S) masuk ke pin *Arduino*
 X8-4 → Masuk ke pin *Arduino* VCC
 X9-1 → Masuk ke pin *Arduino* GND
 X9-2 → *Output* data I (T) masuk ke pin *Arduino*
 X9-3 → *Output* data V (T) masuk ke pin *Arduino*
 X9-4 → Masuk ke pin *Arduino* VCC

3.2.4 Mapping Pin Amphenol



Gambar 3.5 Mapping Pin Amphenol.

Jika dilihat Pada Gambar 3.5 terdapat 2 amphenol, yang pada masing-masing amphenol memiliki 20 pin. Dari gambar tersebut diperoleh keterangan seperti berikut :

- Amphenol A

- Pin 1 → *Output PMT 3 Close*
- Pin 2 → *Output PMT 2 Close*
- Pin 3 → *Push Button Close 1*
- Pin 4 → *Output PMT 1 Close*
- Pin 5 → *Output PMT 3 Open*
- Pin 6 → *Output PMT 2 Open*
- Pin 7 → *Push Button Close 2*
- Pin 8 → *Output PMT 1 Open*
- Pin 9 → *Input sumber 5V*
- Pin 10 → *Push Button Close 3*
- Pin 11 → *Input sumber 5V*
- Pin 12 → *Push Button Open 1*
- Pin 13 → *Input sumber 5V*
- Pin 14 → *Push Button Open 2*
- Pin 15 → *GND Arduino*
- Pin 16 → *Push Button Open 3*
- Pin 17 → *Input sumber 24 V*
- Pin 18 → *Input sumber 24 V*
- Pin 19 → *GND*
- Pin 20 → *GND*

- Amphenol B

- Pin 1 → *Status fasa S*
- Pin 2 → *Status fasa R*
- Pin 4 → *Status fasa T*
- Pin 7 → *Output arus pada fasa R*
- Pin 9 → *Output arus pada fasa S*
- Pin 11 → *Output arus pada fasa T*
- Pin 13 → *Output tegangan pada fasa R*
- Pin 15 → *Output tegangan pada fasa S*
- Pin 17 → *Output tegangan pada fasa T*
- Pin 19 → *GND*
- Pin 20 → *GND*

3.2.5 Modul Sensor Arus ACS712 [10]

Untuk mengukur besaran arus yang melewati suatu kabel atau konduktor diperlukan suatu sensor arus. Pada kenyataan yang digunakan di gardu induk, sensor yang digunakan adalah *current transformer* (CT).

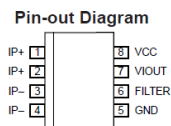
Sedangkan Sensor arus yang digunakan pada pembuatan alat ini adalah modul ACS712-5A. Yang merupakan sensor untuk arus kecil dengan kemampuan sampai 5 A.

Sensor ini digunakan untuk mengukur arus AC yang melalui *incoming* simulasi gardu induk. Karena arus yang diukur merupakan arus AC dan *output* tegangan sensor juga AC.

ACS712 adalah Hall Effect *current sensor*. *Hall effect* allegro ACS712 merupakan sensor yang presisi sebagai sensor arus AC atau DC dalam pembacaan arus didalam dunia industri, otomotif, komersil dan sistem-sistem komunikasi. Pada umumnya aplikasi sensor ini biasanya digunakan untuk mengontrol motor, deteksi beban listrik, *switched-mode power supplies* dan proteksi beban berlebih.

Sensor ini memiliki pembacaan dengan ketepatan yang tinggi, karena didalamnya terdapat rangkaian *low-offset linear Hall* dengan satu lintasan yang terbuat dari tembaga. cara kerja sensor ini adalah arus yang dibaca mengalir melalui kabel tembaga yang terdapat didalamnya yang menghasilkan medan magnet yang di tangkap oleh *integrated* Hall IC dan diubah menjadi tegangan proporsional.

Ketelitian dalam pembacaan sensor dioptimalkan dengan cara pemasangan komponen yang ada didalamnya antara penghantar yang menghasilkan medan magnet dengan *hall transducer* secara berdekatan. Persisnya, tegangan proporsional yang rendah akan menstabilkan Bi CMOS Hall IC yang didalamnya yang telah dibuat untuk ketelitian yang tinggi oleh pabrik. Pada Gambar 3.6 adalah gambar dari *pin out* sensor ACS712 dan Pada Tabel 3.1 adalah *Data Sheet* sensor ACS712.



Gambar 3.6 *Pin Out Diagram* Sensor ACS712.

Tabel 3.1 *Data Sheet* Sensor ACS712.

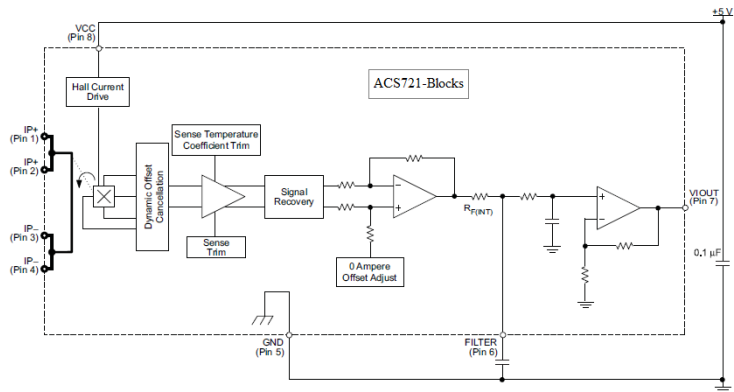
<i>Number</i>	<i>Name</i>	<i>Description</i>
<i>1 and 2</i>	<i>IP+</i>	<i>Terminals for current being; fused internally</i>
<i>3 and 4</i>	<i>IP-</i>	<i>Terminals for current being; fused internally</i>
<i>5</i>	<i>GND</i>	<i>Signal Ground Terminal</i>
<i>6</i>	<i>Filter</i>	<i>Terminal for external capacitor that sets bandwidth</i>
<i>7</i>	<i>VIOut</i>	<i>Analog output signal</i>
<i>8</i>	<i>VCC</i>	<i>Device power supply terminal</i>

Output/keluaran dari sensor ini sebesar ($>VIOUT(Q)$) saat peningkatan arus pada penghantar arus (dari pin 1 dan pin 2 ke pin 3 dan 4), yang digunakan untuk pendeteksian atau perasa arus. Hambatan dalam penghantar sensor sebesar $1,2\text{ m}\Omega$ dengan daya yang rendah. Jalur terminal konduktif secara kelistrikan diisolasi dari sensor *leads*/mengarah (pin 5 sampai pin 8).

Hal ini menjadikan sensor arus ACS712 dapat digunakan pada aplikasi-aplikasi yang membutuhkan isolasi listrik tanpa menggunakan opto-isolator atau teknik isolasi lainnya yang mahal. Ketebalan penghantar arus didalam sensor sebesar 3x kondisi *overcurrent*. Sensor ini telah dikalibrasi oleh pabrik. Pada gambar 3.7 adalah skematik modul sensor arus ACS712-5A.

Beberapa fitur penting dari sensor arus ACS712 adalah:

- Jalur sinyal analog yang rendah *noise*.
- Bandwidth* perangkat diatur melalui pin FILTER yang baru.
- Waktu naik keluaran 5 mikrodetik dalam menanggapi langkah masukan aktif.
- Bandwidth* 50 kHz.
- Total *error* keluaran 1,5% pada $TA = 25^\circ$, dan 4% pada -40° C sampai 85° C .
- Bentuk yang kecil, paket SOIC8 yang kompak.
- Resistansi internal $1,2\text{ m}\Omega$.
- 2,1 kVRMS tegangan isolasi minimum dari pin 1-4 ke pin 5-8.
- Operasi catu daya tunggal 5,0 V.
- Sensitivitas keluaran 66-185 mV/A.
- Tegangan keluaran sebanding dengan arus AC atau DC.



Gambar 3.7 Skematik Modul ACS712-5A.

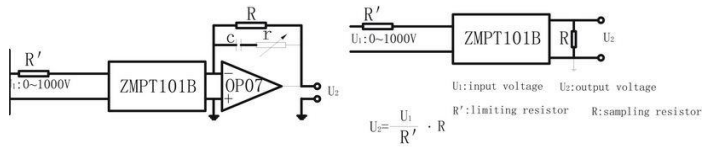
- l. Akurasi sudah diatur oleh pabrik.
 - m. Tegangan *offset* yang sangat stabil.
 - n. Histeresis *magnetic* hampir mendekati nol.
- Keluaran *ratiometric* diambil dari sumber daya.

3.2.6 Modul Sensor Tegangan ZMPT101B [11]

Sensor ZMPT101B merupakan modul sensor tegangan AC yang menggunakan trafo isolasi dengan rasio tegangan 1:1. Pada modul sensor ini sudah terdapat rangkaian summing amplifier. Dengan adanya rangkaian summing amplifier ini dapat menaikkan tegangan AC.

Pada panel kontrol gardu induk, digunakan modul sensor tegangan ZMPT101B 5A untuk mengukur tegangan AC yang melalui *incoming* simulasi gardu induk. Sama halnya dengan modul sensor arus, modul sensor tegangan ini sudah menghasilkan *output* tegangan DC.

Modul ini dilengkapi dengan *transformator step down* AC to AC 2 mA tipe ZMPT101B dan rangkaian komparator untuk mengubah besar tegangan AC menjadi lebih rendah sehingga dapat dihubungkan pada dengan pin *analog to digital* (ADC) *Arduino*. Pada Gambar 3.8 adalah skematik modul sensor tegangan.



Gambar 3.8 Skematik Modul ZMPT101B.

3.3 Perancangan Perangkat Lunak

Untuk menjalankan fungsi-fungsi pada alat ini maka perangkat keras harus dikoordinasikan dengan perangkat lunak. Berikut adalah rancangan perangkat lunak untuk menjalankan peralatan monitoring pengaman beban lebih panel kontrol gardu induk, yang terdiri dari perancangan program di *Arduino* dan tampilan LCD.

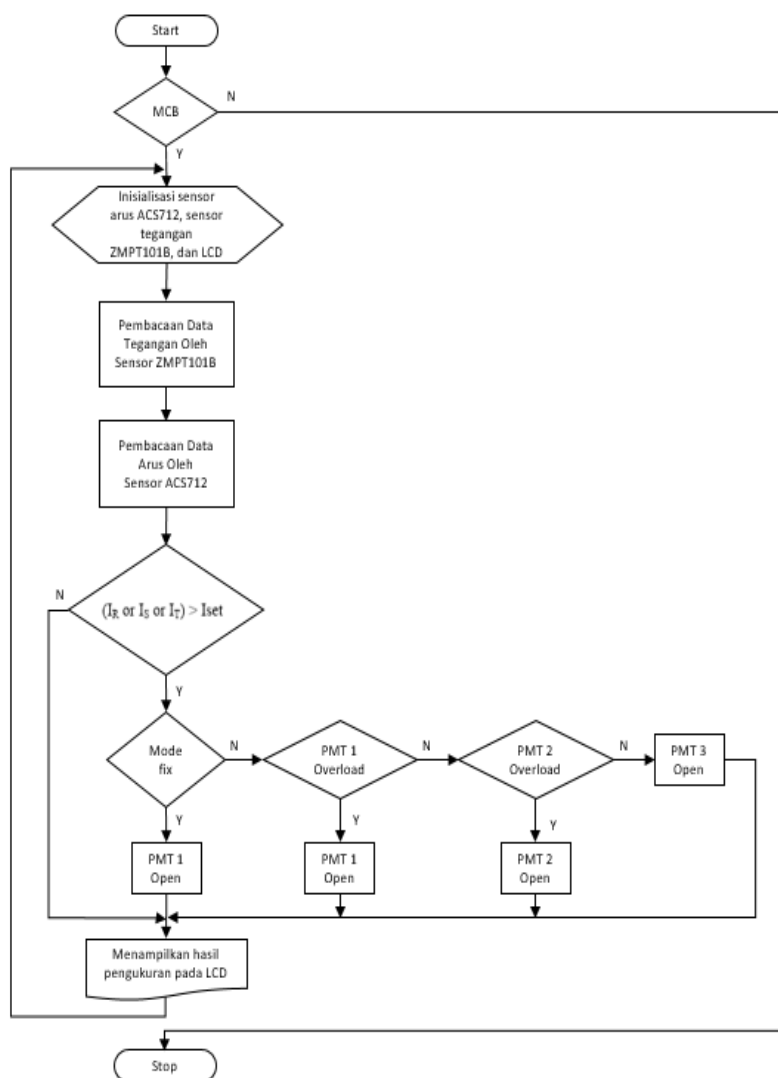
3.3.1 Program Pada *Arduino*

Software yang digunakan untuk melakukan pemrograman terhadap *Arduino* adalah *Arduino IDE*, yang mana pemrograman *Arduino* yang dilakukan pada Tugas Akhir ini menggunakan bahasa C.

Ada beberapa program utama yang akan dibuat untuk menjalankan peralatan pada tugas akhir ini. Untuk memperjelas proses kerja dari alat ini berikut adalah algoritma pemrogramannya :

1. Alat dijalankan dengan memberi masukan tegangan 3 fasa R-S-T 220 V AC atau juga bisa menggunakan 1 fasa 220 V AC. Untuk sistem kontrol PMT diberi tegangan 24 V DC sedangkan untuk sensor di *supply* dengan tegangan 5 V DC.
2. Ketika alat bekerja PMT dimasukkan (*close*) kemudian beban lampu dinyalakan. Maka sensor ACS712 akan membaca Arus R-S-T yang melewati bagian *incoming* alat (sesuai Gambar 3.5).
3. Nilai pembacaan arus dan tegangan akan diolah oleh *Arduino*.
4. Nilai *overload* arus yang dapat diterima alat diberi batasan maksimal, karena pada tiap sensor maksimal mendapatkan beban 2,6 A. Jadi batasan yang diberikan pada tiap-tiap sensor yaitu 2 A. Apabila beban melebihi dari 2 A, maka akan ada salah satu penyulang yang diputus.
5. Hasil pengukuran dari sensor arus dan sensor tegangan ditampilkan pada LCD.

Untuk lebih mempermudah penjelasan maka algoritma di atas dapat diterjemahkan kedalam bentuk *flowchart* Pada Gambar 3.9.



Gambar 3.9 *Flowchart* Kerja Alat.

Penjabaran program utama pada tugas akhir ini adalah sebagai berikut :

a. Program *Arduino* Inisialisasi pin

```
#define S1 37
#define S2 39
#define S3 41

#define St_A 34
#define St_B 49
#define St_C 53

//menuju relay
#define out_close1 43
#define out_close2 45
#define out_close3 47

#define out_open1 23
#define out_open2 25
#define out_open3 27

//dari button
#define in_close3 22
#define in_close2 24
#define in_close1 26

#define in_open3 30
#define in_open2 32
#define in_open1 28

#define line_A 0
#define line_B 1
#define line_C 2

#define line_open 0
#define line_close 1

#define A10 10
#define A11 11
#define A12 12
#define A13 13
#define A14 14
#define A15 15

#define current_treshold 2.0
```

Gambar 3.10 Inisialisasi Program.

Pada Gambar 3.10 terlihat bahwa inisialisasi pin input output pada *Arduino*. Dari gambar tersebut diperoleh keterangan seperti berikut :

- *#define S1*, *#define S2*, dan *#define S3* mendeskripsikan tentang *switch selector* untuk memindah tampilan pada LCD.
- *#define St_A*, *#define St_B*, dan *#define St_C* mendeskripsikan tentang status dari ketiga penyulang baik dalam kondisi *open* maupun *close*.
- *#define out_close1*, *#define out_close2*, dan *#define out_close3* mendeskripsikan tentang relay dalam status *close*.
- *#define out_open1*, *#define out_open2*, dan *#define out_open3* mendeskripsikan tentang relay dalam status *open*.
- *#define in_close1*, *#define in_close2*, dan *#define in_close3* mendeskripsikan tentang *push button close*.
- *#define in_open1*, *#define in_open2*, dan *#define in_open3* mendeskripsikan tentang *push button open*.
- *#define line_A*, *#define line_B*, dan *#define line_C* mendeskripsikan tentang tiga penyulang yang ada panel kontrol.
- *#define A10*, *#define A11*, *#define A12*, *#define A13*, *#define A14*, dan *#define A15* mendeskripsikan tentang pembacaan data dari sensor arus dan sensor tegangan.

b. Program *Arduino Void Setup*

Pada Gambar 3.11 di bawah terlihat bahwa *void setup* pada *Arduino*. Dari gambar tersebut diperoleh keterangan seperti berikut :

- *pinMode S1*, *pinMode S2*, dan *pinMode S3* diaktifkan sebagai *input*, dengan keadaan awal *high*.
- *pinMode St_A*, *pinMode St_B*, dan *pinMode St_C* diaktifkan sebagai *input*, dengan keadaan awal *high*.
- *pinMode in_close1*, *pinMode in_close2*, *pinMode in_close3*, *pinMode in_open1*, *pinMode in_open2*, dan *pinMode in_open3* diaktifkan sebagai *input*, dengan keadaan awal *high*.
- *pinMode out_close1*, *pinMode out_close2*, *pinMode out_close3*, *pinMode out_open1*, *pinMode out_open2*, dan *pinMode out_open3* diaktifkan sebagai *output*.
- *#define line_A*, *#define line_B*, dan *#define line_C* mendeskripsikan tentang tiga penyulang yang ada panel kontrol.

- *#define* A10, *#define* A11, *#define* A12, *#define* A13, *#define* A14, dan *#define* A15 mendeskripsikan tentang pembacaan data dari sensor arus dan sensor tegangan.

```

void out_relay(char line, char state);
void out_relay(char line, char state)
void setup()
{

    //noInterrupts();
    //interrupts();

    // Open serial communications and wait for port to open:
    Serial.begin(115200);
    while (!Serial);
    // wait for serial port to connect. Needed for native USB port only
    Serial.print("R   ");
    pinMode(LED_BUILTIN, OUTPUT);

    pinMode(S1, INPUT);digitalWrite(S1, HIGH);
    pinMode(S2, INPUT);digitalWrite(S2, HIGH);
    pinMode(S3, INPUT);digitalWrite(S3, HIGH);

    pinMode(St_A, INPUT);digitalWrite(St_A, HIGH);
    pinMode(St_B, INPUT);digitalWrite(St_B, HIGH);
    pinMode(St_C, INPUT);digitalWrite(St_C, HIGH);

    pinMode(in_close1, INPUT);digitalWrite(in_close1, HIGH);
    pinMode(in_close2, INPUT);digitalWrite(in_close2, HIGH);
    pinMode(in_close3, INPUT);digitalWrite(in_close3, HIGH);
    pinMode(in_open1, INPUT);digitalWrite(in_open1, HIGH);
    pinMode(in_open2, INPUT);digitalWrite(in_open2, HIGH);
    pinMode(in_open3, INPUT);digitalWrite(in_open3, HIGH);

    pinMode(out_close1, OUTPUT);
    pinMode(out_close2, OUTPUT);
    pinMode(out_close3, OUTPUT);
    pinMode(out_open1, OUTPUT);
    pinMode(out_open2, OUTPUT);
    pinMode(out_open3, OUTPUT);

    Serial.println("new R   ");
    lcd.begin(16,4);
    Serial.println("R   ");
    //start the Ethernet connection and the server:
    Ethernet.begin(mac, ip);
    Serial.println("R   ");
    server.begin();
    Serial.print("server is at ");
    Serial.println(Ethernet.localIP());

    out_relay(line_A,line_close);
    out_relay(line_B,line_close);
    out_relay(line_C,line_close);
}

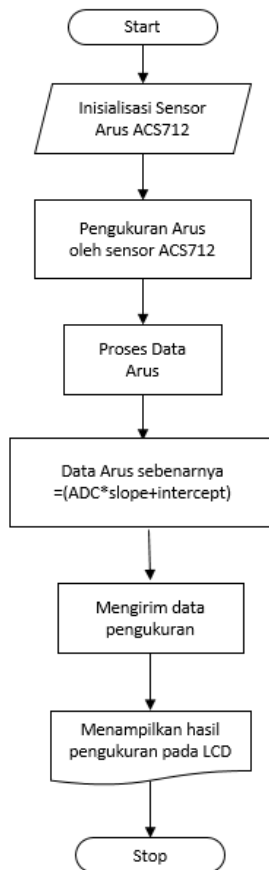
```

Gambar 3.11 Program Arduino Void Setup.

3.3.2 Pemrograman pada Sensor Arus

Pada pemrograman sensor arus ACS712 diawali dengan inisialisasi sensor arus dengan cara diberi masukan beban, kemudian diukur dan diproses datanya. Dari data arus sebenarnya dimasukkan rumus $=(ADC*Slope+Intercept)$, kemudian data ditampilkan pada LCD.

Untuk lebih mempermudah penjelasan di atas dapat diterjemahkan kedalam bentuk *flowchart* Pada Gambar 3.12.

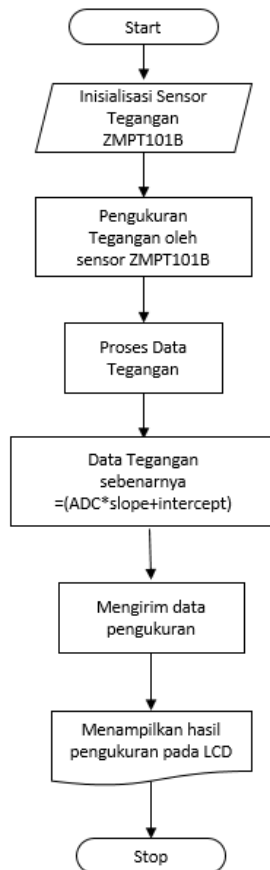


Gambar 3.12 *Flowchart* Pemrograman Sensor Arus.

3.3.3 Pemrograman pada Sensor Tegangan

Pada pemrograman sensor arus ZMPT101B diawali dengan inialisasi sensor tegangan dengan cara diberi masukkan sumber, kemudian diukur dan diproses datanya. Dari data tegangan sebenarnya dimasukkan rumus $=(\text{ADC} * \text{Slope} + \text{Intercept})$, kemudian data ditampilkan pada LCD.

Untuk lebih mempermudah penjelasan di atas dapat diterjemahkan kedalam bentuk *flowchart* Pada Gambar 3.13.



Gambar 3.13 *Flowchart* Pemrograman Sensor Tegangan.

BAB IV

PENGUJIAN DAN ANALISA DATA

Dalam perencanaan dan pembuatan suatu sistem, pengujian dan analisa sangatlah diperlukan, karena dengan pengujian dan analisa dimaksudkan untuk mengetahui kinerja dari sistem dan komponen pendukung yang dibuat sudah sesuai dengan yang direncanakan atau belum.

Pada bab ini akan dibahas tentang pengujian dan analisa baik itu pada *hardware* dan *software* yang mendukung dari dibuatnya tugas akhir ini serta sistem secara keseluruhan.

4.1 Pengujian Hardware

Setelah semua peralatan selesai dibuat sesuai dengan perancangan pada Bab III, maka sebelum semua peralatan tersebut dikoneksikan dan diintegrasikan dengan *software*, perlu dilakukan uji terlebih dahulu performa dari tiap peralatan tersebut.

Pengujian *hardware* yang dilakukan pada bagian ini antara lain pengujian PMT & Panel kontrol *open-close*, Pengujian sensor Arus ACS712, Pengujian sensor Tegangan ZMPT101B, dan LCD.

Pada Gambar 4.1 adalah menunjukkan hasil secara keseluruhan dari perancangan alat pada tugas akhir ini.



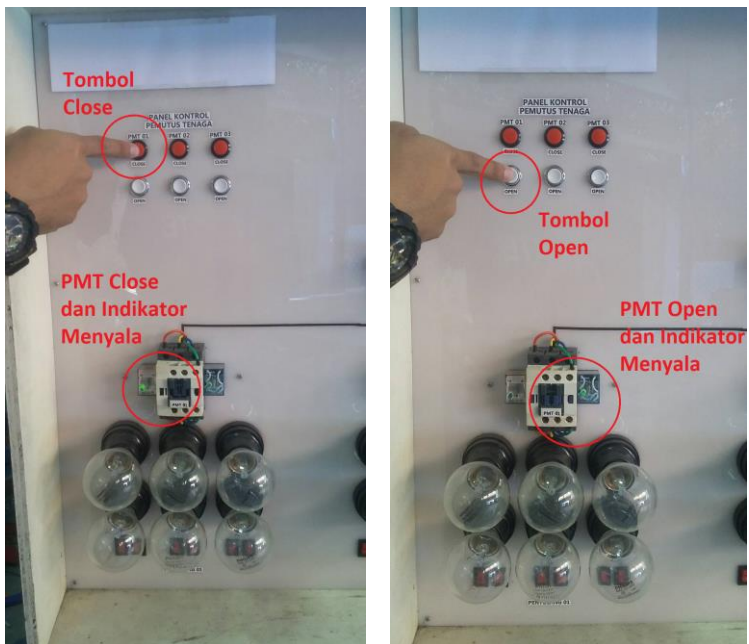
Gambar 4.1 *Hardware Keseluruhan.*

4.1.1 Pengujian PMT dan Panel Kontrol *Open-Close*

Pengujian PMT dan panel kontrol *Open-Close* ini bertujuan untuk mengetahui performa dari rancangan PMT serta panel kontrolnya apakah sudah sesuai dengan yang diharapkan atau belum. Pengujian ini dilakukan dengan cara sebagai berikut :

- Mula-mula alat dinyalakan dengan diberi sumber.
- Kemudian MCB dinyalakan.
- Setelah itu menekan tombol *open* atau tombol *close* pada panel kontrol dan melihat status PMT apakah *open* atau *close*.

Pada Gambar 4.2 ditunjukkan cara pengujiannya. Dan Pada Gambar 4.3 berikut merupakan program pengujian PMT dan panel kontrol *Open-Close*.



Gambar 4.2 Pengujian PMT & Panel Kontrol *Open-Close*.


```

void out_relay(char line, char state);
void out_relay(char line, char state)
{
    if(line==line_A)
    {
        if(state==line_open)
        {
            digitalWrite(out_close1, 1);
            digitalWrite(out_open1, 0);
        }
        else if(state==line_close)
        {
            digitalWrite(out_close1, 0);
            digitalWrite(out_open1, 1);
        }
    }
    if(line==line_B)
    {
        if(state==line_open)
        {
            digitalWrite(out_close2, 1);
            digitalWrite(out_open2, 0);
        }
        else if(state==line_close)
        {
            digitalWrite(out_close2, 0);
            digitalWrite(out_open2, 1);
        }
    }
    if(line==line_C)
    {
        if(state==line_open)
        {
            digitalWrite(out_close3, 1);
            digitalWrite(out_open3, 0);
        }
        else if(state==line_close)
        {
            digitalWrite(out_close3, 0);
            digitalWrite(out_open3, 1);
        }
    }
}

```

Gambar 4.3 Program Pengujian PMT dan Panel Kontrol *Open-Close*.

Tabel 4.1 Hasil Pengujian PMT & Panel Kontrol *Open-Close*.

No.PMT	Pengujian			
	<i>Open</i>		<i>Close</i>	
	Hasil	Indikator	Hasil	Indikator
PMT 1	OK	PMT <i>Open</i> & LED Menyala	OK	PMT <i>Close</i> & LED Menyala
PMT 2	OK	PMT <i>Open</i> & LED Menyala	OK	PMT <i>Close</i> & LED Menyala
PMT 3	OK	PMT <i>Open</i> & LED Menyala	OK	PMT <i>Close</i> & LED Menyala

Pada Tabel 4.1 adalah hasil pengujian masing – masing PMT.

Dari hasil Pada Tabel 4.1 di atas dapat dilihat bahwa performa untuk masing – masing PMT sudah baik dan sesuai dengan yang diharapkan.

4.1.2 Pengujian Sensor Arus ACS712

Pada pengujian sensor arus ini dilakukan dengan cara melakukan pengukuran pada Arus *input* pada masing-masing sensor (R-S-T) dengan cara sebagai berikut :

- Mula-mula sensor arus dirangkai sesuai Pada Gambar 3.4.
- Kemudian sensor disambungkan ke beban.
- Beban yang diberikan merupakan beban lampu.
- Kemudian melihat nilai pembacaan pada *serial monitor Arduino* dan melihat nilai arus sebenarnya pada *multimeter*.

Pada Gambar 4.4 menunjukkan proses pengukuran Arus *input* pada sensor ACS712 menggunakan *multimeter*.

Dan Pada Gambar 4.5 menunjukkan program pembacaan *serial monitor* sensor Arus.



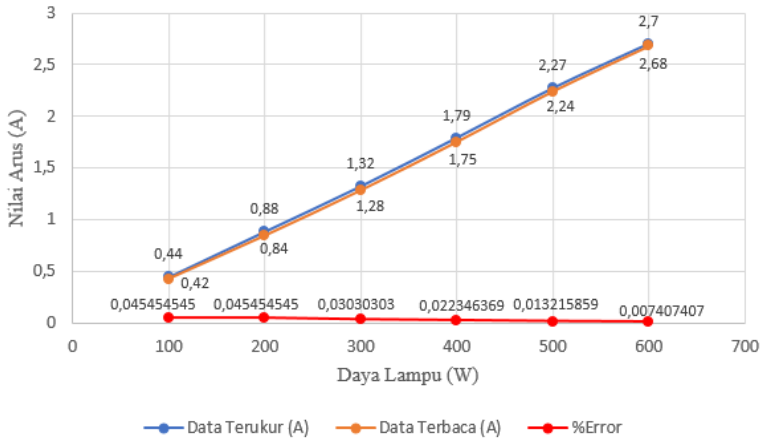
Gambar 4.4 Pengukuran Arus *Input* Sensor ACS712.

```
sensor_Ir = analogRead (A15);  
  
sensor Is = analogRead (A14);  
  
sensor_It = analogRead (A13);
```

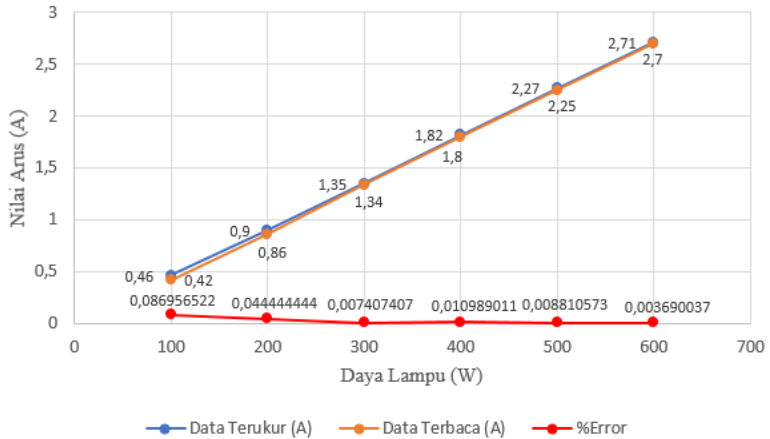
Gambar 4.5 Program Pembacaan *Serial Monitor* Sensor Arus.

Setelah melakukan pengukuran arus *input* sensor dan nilai pembacaan ADC di *serial monitor Arduino*, Pada Gambar 4.6 merupakan grafik hasil pengujian sensor arus pada fasa R, Pada Gambar 4.7 merupakan grafik hasil pengujian sensor arus pada fasa S, dan Pada Gambar 4.8 merupakan grafik hasil pengujian sensor arus pada fasa T.

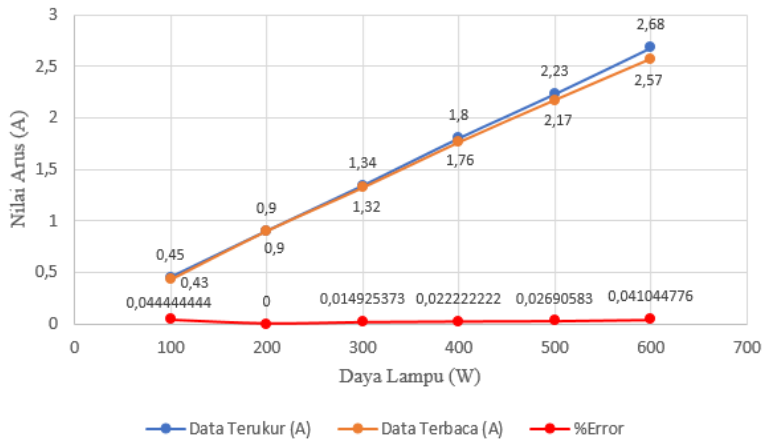
Pada Gambar 4.6, Gambar 4.7, dan Gambar 4.8 dapat dilihat bahwa karakteristik dari masing-masing sensor identik sama tiap fasanya dan memiliki persen eror dibawah 0,1%..



Gambar 4.6 Hasil Pengujian Sensor Arus pada Fasa R.



Gambar 4.7 Hasil Pengujian Sensor Arus pada Fasa S.



Gambar 4.8 Hasil Pengujian Sensor Arus pada Fasa T.

4.1.3 Pengujian Sensor Tegangan ZMPT101B

Pada pengujian sensor tegangan ini dilakukan dengan cara melakukan pengukuran pada Tegangan *output* pada masing-masing sensor (R-S-T) dengan cara sebagai berikut :

- Mula-mula sensor tegangan dirangkai sesuai Pada Gambar 3.4.
- Kemudian sensor mendapat tegangan dari sumber.
- Kemudian melihat nilai pembacaan pada *serial monitor Arduino* dan melihat nilai tegangan sebenarnya pada *multimeter*.

Pada Gambar 4.9 menunjukkan proses pengukuran Tegangan *output* pada sensor ZMPT101B.

Dan Pada Gambar 4.10 menunjukkan program pembacaan *serial monitor* sensor Tegangan.

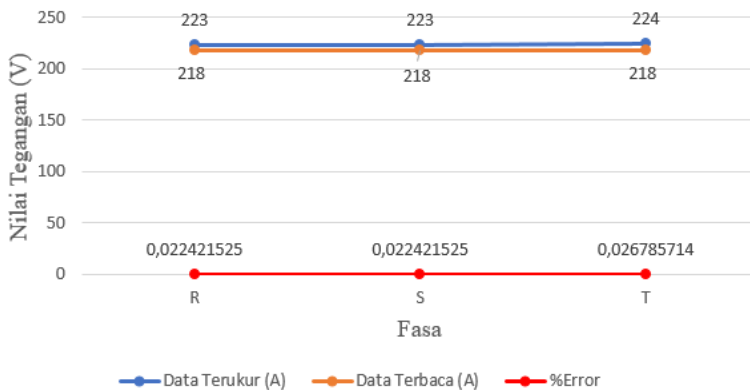


Gambar 4.9 Pengukuran Tegangan *Output* Sensor ZMPT101B.

```
sensor_Vr = analogRead (A12);  
  
sensor_Vs = analogRead (A11);  
  
sensor_Vt = analogRead (A10);
```

Gambar 4.10 Program Pembacaan *Serial Monitor* sensor Tegangan.

Setelah melakukan pengukuran tegangan *output* sensor dan nilai pembacaan ADC di *serial monitor Arduino*, Pada Gambar 4.11 merupakan grafik hasil pengujian sensor arus pada fasa R, S, dan T.



Gambar 4.11 Hasil Pengujian Sensor Tegangan.

4.1.4 Pengujian Tampilan LCD

Pengujian Tampilan LCD ini bertujuan untuk mengetahui tampilan LCD apakah sudah sesuai dengan yang diharapkan atau belum. Pengujian ini dilakukan dengan cara sebagai berikut :

- Mula-mula alat dinyalakan dengan diberi sumber.
- Kemudian MCB dinyalakan.
- Dan mencoba beberapa beban.
- Setelah itu melihat hasil pada tampilan di LCD, apakah sudah sesuai atau belum.

Pada Gambar 4.12 merupakan tampilan pada LCD.



Gambar 4.12 Tampilan pada LCD.

Dari Gambar 4.12 dapat dilihat bahwa tampilan LCD sudah baik dan sesuai dengan yang diharapkan yaitu dapat menampilkan line, nilai arus, nilai tegangan, dan status dari tiap PMT.

4.1.5 Pengujian *Switch Selector*

Pengujian *switch selector* ini bertujuan untuk mengetahui fungsi *switch selector* apakah sudah sesuai dengan yang diharapkan atau belum. Pengujian ini dilakukan dengan cara sebagai berikut :

- Mula-mula alat dinyalakan dengan diberi sumber.
- Kemudian MCB dinyalakan.
- Dan mencoba beberapa beban.
- Setelah itu memputar-putar *switch selector* dan melihat pada tampilan di LCD, apakah sudah sesuai atau belum.

Pada Gambar 4.13 merupakan pengujian *switch selector*..



Gambar 4.13 Pengujian *Switch selector*.

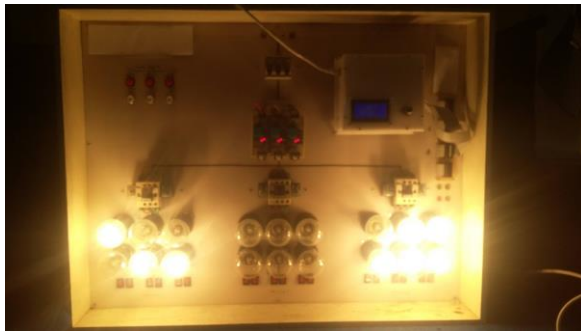
Dari Gambar 4.13 dapat dilihat bahwa fungsi *switch selector* sudah baik dan sesuai dengan yang diharapkan yaitu dapat merubah tampilan pada LCD.

4.2 Pengujian Keseluruhan

Dalam perancangan alat ini setelah dilakukan pengujian tiap komponen maka dilakukan pengujian alat secara keseluruhan. Berupa pengujian dua mode *trip* pada saat alat kelebihan beban. Pada alat ini diberi batasan maksimal arus yang dapat diterima adalah 2A. Ketika beban melebihi 2A maka alat akan melakukan pemutusan pada salah satu penyulang. Pada Gambar 4.14 merupakan gambar hasil pengujian mode *trip fix* dan Pada Gambar 4.15 merupakan gambar hasil pengujian mode *trip rolling*. Perpindahan mode dapat dilakukan dengan telemetring.



Gambar 4.14 Hasil Pengujian Mode *Trip Fix*



Gambar 4.15 Hasil Pengujian Mode *Trip Rolling*

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Dari hasil yang telah didapatkan selama proses pembuatan alat untuk Tugas Akhir ini, maka dapat ditarik beberapa kesimpulan dan saran supaya nantinya bisa memberikan bermanfaat bagi pihak yang bersangkutan.

5.1 Kesimpulan

1. Hasil pengujian sensor arus ACS712 dapat dilihat bahwa sensor arus pada masing masing fasa R, S, dan T memiliki nilai persen eror kurang dari 0,1% pada masing-masing beban. Kecuali pada saat beban lampu 2 pada fasa T memiliki persen eror 0%
2. Hasil pengujian sensor tegangan ZMPT101B dapat dilihat bahwa sensor tegangan pada masing-masing fasa R, S, dan T memiliki nilai persen eror kurang dari 0,1%.

5.2 Saran

Beberapa saran untuk pengembangan alat ini adalah sebagai berikut:

1. Mekanisme monitoring pengamanan seperti pada tugas akhir ini bisa diterapkan pada gardu induk yang sering mengalami *overload* atau gardu induk yang pembebanannya hampir mencapai kapasitas.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Doni Irifan, “Rancang Bangun Simulasi Pengaman Beban Lebih Transformator Gardu Induk Menggunakan *Programmable Logic Control*”, **Tugas Akhir**, Program D3 Teknik Elektro FTI-ITS, Surabaya, 2013.
- [2], **Pengoprasian Peralatan Gardu Induk**, PT.PLN (Persero) Pusat Pendidikan dan Pelatihan, Jakarta, 2009.
- [3], **Trip Rele Incoming Akibat Keterlambatan Sistem Proteksi Pada Penyulang**, PT.PLN (Persero) Area Pengatur Distribusi Surabaya, Surabaya, 2010.
- [4], **Pemeliharaan Trafo Arus(CT) Dan Trafo Tegangan (PT)**, PT.PLN (Persero) Pusat Pendidikan dan Pelatihan, Jakarta, 2009.
- [5], **Sensor Tegangan AC Untuk Arduino**, <http://saptaji.com/2016/11/29/sensor-tegangan-ac-untuk-arduino/>, 26 Mei 2018.
- [6], **Arduino Mega 2560 Mikrokontroler**, <http://www.labelektronika.com/2017/02/arduino-mega-2560-mikrokontroler.html>, 26 Mei 2018.
- [7] Mochamad Fajar Wicaksono dan Hidayat, “**Mudah Belajar Mikrokontroler Arduino**”, Informatika, Bandung, 2017.
- [8], **LCD (Liquid Crystal Display)**, <http://elektronika-dasar.web.id/lcd-liquid-cristal-display/>, 26 Mei 2018.
- [9], **Komponen Elektronika**, <https://teknikelektronika.com/jenis-jenis-komponen-elektronika-beserta-fungsi-dan-simbolnya/>
- [10] Haviz Setiawan, **Sensor Arus efek Hall ACS721 (Hall Effect Allegro ACS712)**, URL: <http://ilmubawang.com/>, 24 Mei 2018.
- [11], **Cara Mengakses Sensor Tegangan 220 V**, <https://www.nyebarilmu.com/cara-mengakses-sensor-tegangan-220v-zmpt101b/>, 26 Mei 2018.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A DATA

A.1 Data Sensor Arus

No	Fasa	Nyala Lampu	P Lampu (W)	I Lampu (A)		ADC	% Eror
				Data Alat Ukur	Data Terbaca		
1	R	1	100	0,44	0,42	533	0,045455
2		2	200	0,88	0,84	556	0,045455
3		3	300	1,32	1,28	581	0,030303
4		4	400	1,79	1,75	606	0,022346
5		5	500	2,27	2,24	633	0,013216
6		6	600	2,7	2,68	655	0,007407
7	S	1	100	0,46	0,42	535	0,086957
8		2	200	0,9	0,86	560	0,044444
9		3	300	1,35	1,34	584	0,007407
10		4	400	1,82	1,8	609	0,010989
11		5	500	2,27	2,25	633	0,008811
12		6	600	2,71	2,7	657	0,00369
13	T	1	100	0,45	0,43	533	0,044444
14		2	200	0,9	0,9	556	0
15		3	300	1,34	1,32	580	0,014925
16		4	400	1,8	1,76	606	0,022222
17		5	500	2,23	2,17	630	0,026906
18		6	600	2,68	2,57	655	0,041045

A.2 Data Sensor Tegangan

No	Fasa	Tegangan Terukur (V)		ADC	% Eror
		Data Alat Ukur	Data Terbaca (LCD)		
1	R	223	218	793	0,022422
2	S	223	218	787	0,022422
3	T	224	218	789	0,026786

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN B PROGRAM

B.1 Program Arduino

```
#include "Arduino.h"  
//The setup function is called once at startup of the sketch  
/*
```

Web Server

A simple web server that shows the value of the analog input pins.
using an Arduino Wiznet Ethernet shield.

Circuit:

- * Ethernet shield attached to pins 10, 11, 12, 13
- * Analog inputs attached to pins A0 through A5 (optional)

created 18 Dec 2009
by David A. Mellis
modified 9 Apr 2012
by Tom Igoe
modified 02 Sept 2015
by Arturo Guadalupi

```
*/
```

```
#include <SPI.h>  
#include <Ethernet.h>  
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>
```

```
#define S1 37  
#define S2 39  
#define S3 41
```

```
#define St_A 34  
#define St_B 49  
#define St_C 53
```

```
#define St_D 34

//menuju relay
#define out_close1 43
#define out_close2 45
#define out_close3 47

#define out_open1 23
#define out_open2 25
#define out_open3 27

//dari button
#define in_close3 22
#define in_close2 24
#define in_close1 26

#define in_open3 30
#define in_open2 32
#define in_open1 28

#define line_A 0
#define line_B 1
#define line_C 2

#define line_open 0
#define line_close 1

#define A10 10
#define A11 11
#define A12 12
#define A13 13
#define A14 14
#define A15 15

#define current_treshold 2.0
```

```
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // Set the
LCD I2C address
```

```
// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
```

```
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
IPAddress ip(192, 168, 1, 177);
EthernetServer server(80);
```

```
char char_buff[100];
```

```
const int analogIn = A0;          // inisialisasi pin A12 sebagai pembacaan
    sensor
float v_Out;
```

```
int head=0;
unsigned char datum0=0,datum1=0,datum2=0;
```

```
float sensor_Vr = 0,sensor_Ir = 0;
float sensor_Vs = 0,sensor_Is = 0;
float sensor_Vt = 0,sensor_It = 0;// setting nilai default pembacaan sensor
    0
```

```
float Ir=0,Vr=0,buff_Vr=0,buff_Ir=0;
float Is=0,Vs=0,buff_Vs=0,buff_Is=0;
float It=0,Vt=0,buff_Vt=0,buff_It=0;
```

```
char state_Vr=0;
char state_Ir=0;
char state_Vs=0;
char state_Is=0;
char state_Vt=0;
char state_It=0;
```

```
int Vint;
```

```

char mode=0,control=0;
char start_milis_delay=0;

int milis_delay=0;

unsigned long previousMillis = 0;      // will store last time LED was
        updated

// constants won't change:
const long interval = 1000;

void out_relay(char line, char state);
void out_relay(char line, char state)
{
    if(line==line_A)
    {
        if(state==line_open)
        {
            digitalWrite(out_close1, 1);//delay(100);
            digitalWrite(out_open1, 0);
        }
        else if(state==line_close)
        {
            digitalWrite(out_close1, 0);//delay(100);
            digitalWrite(out_open1, 1);//delay(5000);
        }
    }
    if(line==line_B)
    {
        if(state==line_open)
        {
            digitalWrite(out_close2, 1);//delay(100);
            digitalWrite(out_open2, 0);//delay(5000);
        }
        else if(state==line_close)
        {
            digitalWrite(out_close2, 0);//delay(100);
            digitalWrite(out_open2, 1);//delay(5000);
        }
    }
}

```

```

    }
}
if(line==line_C)
{
    if(state==line_open)
    {
        digitalWrite(out_close3, 1);//delay(100);
        digitalWrite(out_open3, 0);//delay(5000);
    }
    else if(state==line_close)
    {
        digitalWrite(out_close3, 0);//delay(100);
        digitalWrite(out_open3, 1);//delay(5000);
    }
}
}
uint8_t get_input()
{
    uint8_t input=0;
    char delay_count=20;
    unsigned int hold=0;

    input|=0xc0;

    if(!digitalRead(in_close1))
    {
        while(!digitalRead(in_close1))
        {
            delay(2);
            //Serial.println(hold,DEC);
            hold++;
            if(hold>=delay_count)
            {
                input|=B000010;

                //Serial.println(input,BIN);
                delay(1000);
                break;
            }
        }
    }
}

```

```

    }
}
else if(!digitalRead(in_open1))
{
    while(!digitalRead(in_open1))
    {
        delay(2);
        hold++;
        //Serial.println(hold,DEC);
        if(hold>=delay_count)
        {
            input|=B000011;

            //Serial.println(input,BIN);
            //delay(1000);
            break;
        }
    }
}

if(!digitalRead(in_close2))
{
    while(!digitalRead(in_close2))
    {
        delay(2);
        hold++;
        if(hold>=delay_count)
        {
            input|=B001000;
            //delay(1000);
            break;
        }
    }
}
else if(!digitalRead(in_open2))
{
    while(!digitalRead(in_open2))
    {
        delay(2);

```

```

    hold++;
    if(hold>=delay_count)
    {
        input|=B001100;
        //delay(1000);
        break;
    }
}
}

if(!digitalRead(in_close3))
{
    while(!digitalRead(in_close3))
    {
        delay(2);
        hold++;
        if(hold>=delay_count)
        {
            input|=B100000;
            //delay(1000);
            break;
        }
    }
}
else if(!digitalRead(in_open3))
{
    while(!digitalRead(in_open3))
    {
        delay(2);
        hold++;
        if(hold>=delay_count)
        {
            input|=B110000;
            //delay(1000);
            break;
        }
    }
}
}

```

```

    return input;
}
void debug()
{
    sensor_Vr = analogRead (A12);    // read the analog in value:
    sensor_Ir = analogRead (A15);
    sensor_Vs = analogRead (A11);    // read the analog in value:
    sensor_Is = analogRead (A14);
    sensor_Vt = analogRead (A10);    // read the analog in value:
    sensor_It = analogRead (A13);

    Serial.print(sensor_Vr);
    Serial.print(" ");
    Serial.print(sensor_Ir);
    Serial.print("R ");
    Serial.print(sensor_Vs);
    Serial.print(" ");
    Serial.print(sensor_Is);
    Serial.print("S ");
    Serial.print(sensor_Vt);
    Serial.print(" ");
    Serial.print(sensor_It);
    Serial.print("T ");

    Serial.print(digitalRead(in_close1));
    Serial.print(digitalRead(in_open1));
    Serial.print(" ");
    Serial.print(digitalRead(in_close2));
    Serial.print(digitalRead(in_open2));
    Serial.print(" ");
    Serial.print(digitalRead(in_close3));
    Serial.print(digitalRead(in_open3));
    Serial.print(" ");
    Serial.print(digitalRead(St_D));
    Serial.print(digitalRead(St_B));
    Serial.print(digitalRead(St_C));
    Serial.print(" ");

    Serial.println(get_input(),BIN);

```



```

}

void send2web(uint8_t refresh_rate)
{
    EthernetClient client = server.available();
    if (client) {
        Serial.println("new client");
        // an http request ends with a blank line
        boolean currentLineIsBlank = true;
        while (client.connected())
        {
            if (client.available()) {
                char c = client.read();
                Serial.write(c);

                if(c=='#')head=1;
                else if(head==1 && (c=='0' || c=='1'))
                {
                    if(control)
                    {
                        if(c=='0')out_relay(line_A,line_close);
                        else if(c=='1')out_relay(line_A,line_open);

                    }
                    head=2;
                }
                else if(head==2 && (c=='0' || c=='1'))
                {
                    if(control)
                    {
                        if(c=='0')out_relay(line_B,line_close);
                        else if(c=='1')out_relay(line_B,line_open);

                    }
                    head=3;
                }
                else if(head==3 && (c=='0' || c=='1'))
                {
                    if(control)

```

```

{
    if(c=='0')out_relay(line_C,line_close);
    else if(c=='1')out_relay(line_C,line_open);

}
head=4;
}
else if(head==4 && (c=='0' || c=='1'))
{
    if(c=='0')control=0;
    else if(c=='1')control=1;
    head=5;
}
else if(head==5 && (c=='0' || c=='1'))
{
    if(c=='0')mode=0;
    else if(c=='1')mode=1;
    head=0;
}

// if you've gotten to the end of the line (received a newline
// character) and the line is blank, the http request has ended,
// so you can send a reply
if (c == '\n' )
{
    // send a standard http response header
    sprintf(char_buff,"Refresh: %d",refresh_rate);
    Serial.println(char_buff);

    //client.println("HTTP/1.1 200 OK");
    //client.println("Content-Type: text/html");
    //client.println("Connection: close"); // the connection will be
    //closed after completion of the response
    //client.println("Refresh: 5"); // refresh the page automatically
    //every 5 sec
    //client.println(char_buff);
    //client.println();
    //client.println("<!DOCTYPE HTML>");

```

```

//client.println("<html>");
// output the value of each analog input pin
//for (int analogChannel = 10; analogChannel <= 15;
  analogChannel++) {
  //int sensorReading = analogRead(analogChannel);
int Isend=0;

  Vint=(int) Vr;
  Isend=Ir*1000;
  //Ir*=1000;
  client.print("Vr=");
  client.print(Vint);
  client.print("V");
  client.print(" Ir=");
  client.print(Isend);
  client.println("A");
  //client.println("<br />");

  Vint=(int) Vs;
  Isend=Is*1000;
  //Is*=1000;
  client.print("Vs=");
  client.print(Vint);
  client.print("V");
  client.print(" Is=");
  client.print(Isend);
  client.println("A");
  //client.println("<br />");

  Vint=(int) Vt;
  Isend=It*1000;
  //It*=1000;
  client.print("Vt=");
  client.print(Vint);
  client.print("V");
  client.print(" It=");
  client.print(Isend);
  client.println("A");

```

```

    client.print('=');
    client.print(digitalRead(St_D));
    client.print('=');
    client.print(digitalRead(St_B));
    client.print('=');
    client.println(digitalRead(St_C));

    //client.println("<br />");

    //client.print(sensorReading);
    //client.println("<br />");

    //client.println("</html>");
    break;
}
if (c == '\n') {
    // you're starting a new line
    currentLineIsBlank = true;
} else if (c != '\r') {
    // you've gotten a character on the current line
    currentLineIsBlank = false;
}
}
}
// give the web browser time to receive the data
delay(1);
// close the connection:
client.stop();
Serial.println("client disconnected");
}
}

void setup()
{

    //noInterrupts();
    //interrupts();

    // Open serial communications and wait for port to open:

```

```

Serial.begin(115200);
while (!Serial);
// wait for serial port to connect. Needed for native USB port only
Serial.print("R  ");
pinMode(LED_BUILTIN, OUTPUT);

pinMode(S1, INPUT);digitalWrite(S1, HIGH);
pinMode(S2, INPUT);digitalWrite(S2, HIGH);
pinMode(S3, INPUT);digitalWrite(S3, HIGH);

pinMode(St_A, INPUT);digitalWrite(St_A, HIGH);
pinMode(St_B, INPUT);digitalWrite(St_B, HIGH);
pinMode(St_C, INPUT);digitalWrite(St_C, HIGH);
pinMode(St_D, INPUT);digitalWrite(St_D, HIGH);

pinMode(in_close1, INPUT);digitalWrite(in_close1, HIGH);
pinMode(in_close2, INPUT);digitalWrite(in_close2, HIGH);
pinMode(in_close3, INPUT);digitalWrite(in_close3, HIGH);
pinMode(in_open1, INPUT);digitalWrite(in_open1, HIGH);
pinMode(in_open2, INPUT);digitalWrite(in_open2, HIGH);
pinMode(in_open3, INPUT);digitalWrite(in_open3, HIGH);

pinMode(out_close1, OUTPUT);
pinMode(out_close2, OUTPUT);
pinMode(out_close3, OUTPUT);
pinMode(out_open1, OUTPUT);
pinMode(out_open2, OUTPUT);
pinMode(out_open3, OUTPUT);

Serial.println("new R  ");
lcd.begin(16,4);
Serial.println("R  ");
//start the Ethernet connection and the server:
Ethernet.begin(mac, ip);
Serial.println("R  ");
server.begin();
Serial.print("server is at ");
Serial.println(Ethernet.localIP());

```

```

    out_relay(line_A,line_close);
    out_relay(line_B,line_close);
    out_relay(line_C,line_close);
}

void loop()
{
    send2web(1);

    if(!control)
    {
        if (get_input()==B11000010)out_relay(line_C,line_close);
        else if(get_input()==B11000011)out_relay(line_C,line_open);
        else if(get_input()==B11001000)out_relay(line_B,line_close);
        else if(get_input()==B11001100)out_relay(line_B,line_open);
        else if(get_input()==B11100000)out_relay(line_A,line_close);
        else if(get_input()==B11110000)out_relay(line_A,line_open);

    }
    debug();

    /*int ADC_0=analogRead(0); // baca adc (0-1023)
    int ADC_1=analogRead(1); // baca adc (0-1023)
    int ADC_2=analogRead(2); // baca adc (0-1023)
    int ADC_3=analogRead(3); // baca adc (0-1023)
    int ADC_4=analogRead(4); // baca adc (0-1023)
    int ADC_5=analogRead(5); // baca adc (0-1023)
    Serial.write(highByte(ADC_0));//kirim 8 bit atas
    Serial.write(lowByte(ADC_0));//kirim 8 bit bawah
    Serial.write(highByte(ADC_1));//kirim 8 bit atas
    Serial.write(lowByte(ADC_1));//kirim 8 bit bawah
    Serial.write(highByte(ADC_2));//kirim 8 bit atas
    Serial.write(lowByte(ADC_2));//kirim 8 bit bawah
    Serial.write(highByte(ADC_3));//kirim 8 bit atas
    Serial.write(lowByte(ADC_3));//kirim 8 bit bawah
    Serial.write(highByte(ADC_4));//kirim 8 bit atas
    Serial.write(lowByte(ADC_4));//kirim 8 bit bawah
    Serial.write(highByte(ADC_5));//kirim 8 bit atas

```

```

Serial.write(lowByte(ADC_5)); // kirim 8 bit bawah
//=====
delay(150);
*/

unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval)
{
    // save the last time you blinked the LED
    previousMillis = currentMillis;

    // if the LED is off turn it on and vice-versa:
    digitalWrite(LED_BUILTIN, digitalRead(LED_BUILTIN) ^ 1);
    if(start_milis_delay)milis_delay++;
    else milis_delay=0;

    if(mode==1 && control==0 && (state_Ir==2 || state_Is==2 ||
        state_It==2))
    {
        start_milis_delay=1;
        if(milis_delay<1)
        {
            out_relay(line_A,line_open);
            out_relay(line_B,line_close);
            out_relay(line_C,line_close);
        }
        else if(milis_delay<4)
        {
            out_relay(line_A,line_close);
            out_relay(line_B,line_open);
            out_relay(line_C,line_close);
        }
        else if(milis_delay<7)
        {
            out_relay(line_A,line_close);
            out_relay(line_B,line_close);
            out_relay(line_C,line_open);
            milis_delay=0;
        }
    }
}

```

```

    }
    //else if(milis_delay<8) milis_delay=0;

}
else if (mode==0 && control==0 && (state_Ir==2 || state_Is==2 ||
    state_It==2))
{
    out_relay(line_A,line_open);
}
else if(state_Ir<=1 && state_Is<=1 && state_It<=1)
{
    //out_relay(line_A,line_close);
    //out_relay(line_B,line_close);
    //out_relay(line_C,line_close);
    start_milis_delay=0;
}

/*if(milis_delay<2)
{
    //digitalWrite(out_close3, 0);
    //digitalWrite(out_open3, 1);//delay(5000);

    //digitalWrite(out_open1, 0);//delay(100);
    //digitalWrite(out_close1, 1);//delay(5000);
    out_relay(line_R,line_close);
    out_relay(line_S,line_close);
    out_relay(line_T,line_close);
}
else if(milis_delay<3)
{
    //digitalWrite(out_close1, 0);
    //digitalWrite(out_open1, 1);//delay(5000);

    //digitalWrite(out_open2, 0);//delay(100);
    //digitalWrite(out_close2, 1);//delay(5000);
    out_relay(line_R,line_close);
    out_relay(line_S,line_close);
    out_relay(line_T,line_close);
}

```



```

}
else if(milis_delay<4)
{
  //digitalWrite(out_close2, 0);
  //digitalWrite(out_open2, 1);//delay(5000);

  //digitalWrite(out_open3, 0);//delay(100);
  //digitalWrite(out_close3, 1);//delay(5000);
  out_relay(line_R,line_close);
  out_relay(line_S,line_close);
  out_relay(line_T,line_close);

  lcd.clear();
  milis_delay=0;
}*/

if(!control)
{
  lcd.backlight();

  if(!digitalRead(S1))
  {
    Vint=(int) Vr;

    lcd.setCursor(5,0);
    lcd.print("Line R");

    lcd.setCursor(2,1);
    lcd.print("Vr=");
    lcd.print(Vint);
    lcd.print("V   ");

    lcd.setCursor(2,2);
    lcd.print("Ir=");
    lcd.print(Ir);
    lcd.print("A   ");

    if(digitalRead(St_A) || digitalRead(St_B) || digitalRead(St_C)){ }
    else

```

```

{
  if(state_Vr==1)
  {
    lcd.setCursor(2,3);
    lcd.print("Online  ");
  }
  else if(state_Vr==0)
  {
    lcd.setCursor(2,3);
    lcd.print("Offline  ");
  }
  else if(state_Vr==2)
  {
    lcd.setCursor(2,3);
    lcd.print("Over voltage");
  }
  else if(state_Vr==3)
  {
    lcd.setCursor(2,3);
    lcd.print("Under voltage");
  }
  if(state_Ir==2)
  {
    lcd.setCursor(2,3);
    lcd.print("Over Current ");
  }
}
}
else if(!digitalRead(S2))
{
  Vint=(int) Vs;

  lcd.setCursor(5,0);
  lcd.print("Line S");

  lcd.setCursor(2,1);
  lcd.print("Vs=");
  lcd.print(Vint);
  lcd.print("V  ");
}

```

```

lcd.setCursor(2,2);
//zlcd.clear();
lcd.print("Is=");
lcd.print(Is);
lcd.print("A  ");

if(digitalRead(St_A) || digitalRead(St_B) || digitalRead(St_C)){ }
else
{
  if(state_Vs==1)
  {
    lcd.setCursor(2,3);
    lcd.print("Normal  ");
  }
  else if(state_Vs==0)
  {
    lcd.setCursor(2,3);
    lcd.print("Offline  ");
  }
  else if(state_Vs==2)
  {
    lcd.setCursor(2,3);
    lcd.print("Over voltage");
  }
  else if(state_Vs==3)
  {
    lcd.setCursor(2,3);
    lcd.print("Under voltage");
  }

  if(state_Is==2)
  {
    lcd.setCursor(2,3);
    lcd.print("Over Current ");
  }
}
}
else if(!digitalRead(S3))

```

```

{
  Vint=(int) Vt;

  lcd.setCursor(5,0);
  lcd.print("Line T");

  lcd.setCursor(2,1);
  lcd.print("Vt=");
  lcd.print(Vint);
  lcd.print("V ");

  lcd.setCursor(2,2);
  lcd.print("It=");
  lcd.print(It);
  lcd.print("A ");

  if(digitalRead(St_A) || digitalRead(St_B) || digitalRead(St_C)){ }
  else
  {
    if(state_Vt==1)
    {
      lcd.setCursor(2,3);
      lcd.print("Normal ");
    }
    else if(state_Vt==0)
    {
      lcd.setCursor(2,3);
      lcd.print("Offline ");
    }
    else if(state_Vt==2)
    {
      lcd.setCursor(2,3);
      lcd.print("Over voltage");
    }
    else if(state_Vt==3)
    {
      lcd.setCursor(2,3);
      lcd.print("Under voltage");
    }
  }
}

```

```

    if(state_It==2)
    {
        lcd.setCursor(2,3);
        lcd.print("Over Current ");
    }
}
}

if(digitalRead(St_A) || digitalRead(St_B) || digitalRead(St_C))
{
    //lcd.setCursor(0,3);
    //lcd.print(" ");
    if(digitalRead(St_A) && digitalRead(St_B) && digitalRead(St_C))
    {
        lcd.setCursor(2,3);
        lcd.print("All");
    }
    else
    {
        if(digitalRead(St_A))
        {
            lcd.setCursor(0,3);
            lcd.print(" A ");
        }else
        {
            lcd.setCursor(0,3);
            lcd.print(" ");
        }
        if(digitalRead(St_B))
        {
            lcd.setCursor(3,3);
            lcd.print("B ");
        }else
        {
            lcd.setCursor(3,3);
            lcd.print(" ");
        }
        if(digitalRead(St_C))

```

```

        {
            lcd.setCursor(4,3);
            lcd.print("C");
        }
    }
    lcd.setCursor(5,3);
    lcd.print(" open");
}
}
else if(control)
{
    lcd.clear();
    lcd.noBacklight();
}
}

if(datum0<=50)
{
    if(buff_Ir<=sensor_Ir)buff_Ir=sensor_Ir;
    if(buff_Vr<=sensor_Vr)buff_Vr=sensor_Vr;

    /*?if(!digitalRead(S1))
    {
        lcd.setCursor(2,2);
        lcd.print(sensor_Vr);
        lcd.setCursor(2,3);
        lcd.print(sensor_Ir);
        //buff_ir=0;
    }*/
    datum0++;
}
else //itung vr
{
    datum0=0;

    if (buff_Vr>675 && buff_Vr<825) //itung Vr
    {
        state_Vr=1;
    }
}

```

```

Vr=0.268*buff_Vr+ 6.57;

if(buff_Ir>504 && buff_Ir<533)Ir=buff_Ir*0.0152 - 7.6469;
    else if(buff_Ir>=533 && buff_Ir<556)Ir=buff_Ir*0.0191 - 9.7565;
    else if(buff_Ir>=556 && buff_Ir<581)Ir=buff_Ir*0.0176 - 8.9056;
    else if(buff_Ir>=581)Ir=buff_Ir*0.0188 - 9.6028;
    else if(buff_Ir==503)Ir=0.00;
    else if(buff_Ir<503)Ir=0;

if(Ir<=0)
{
    Ir=0;
    state_Ir=0;
}
if(Ir<=current_treshold)state_Ir=1;
else if(Ir>current_treshold)state_Ir=2;

buff_Ir=0;
buff_Vr=0;
datum0=0;

}
else if(buff_Vr<550 && buff_Vr>470)
{
    state_Vr=0;
    Vr=0;
    Ir=0;
    buff_Ir=0;
    buff_Vr=0;
    datum0=0;
}
else if(buff_Vr>825)
{
    Vr=0.277*buff_Vr+ 13.57;
    state_Vr=2;
    buff_Ir=0;
    buff_Vr=0;
    //out_relay(line_R,line_open);
}

```

```

else if(buff_Vr<675)
{
    Vr=0.277*buff_Vr+ 13.57;
    state_Vr=4;
    buff_Ir=0;
    buff_Vr=0;
}

} //itung vs

if(datum1<=50)
{
    if(buff_Is<=sensor_Is)buff_Is=sensor_Is;
    if(buff_Vs<=sensor_Vs)buff_Vs=sensor_Vs;

    datum1++;
}
else //itung vs
{
    datum1=0;
    /*if(!digitalRead(S1))
    {
        lcd.setCursor(2,2);
        lcd.print(buff_Vr);
        lcd.setCursor(2,3);
        lcd.print(buff_Ir);
    }*/
    if (buff_Vs>675 && buff_Vs<825)
    {
        state_Vs=1;

        Vs=0.268*buff_Vs+ 6.57;

        if(buff_Is>504 && buff_Is<535)Is=buff_Is*0.0148 - 7.4787;
        else if(buff_Is>=535 && buff_Is<560)Is=buff_Is*0.0176 - 8.956;
        else if(buff_Is>=560 && buff_Is<584)Is=buff_Is*0.0188 - 9.6;
        else if(buff_Is>=584)Is=buff_Is*0.0188 - 9.6292;
        else if(buff_Is==503)Is=0.00;
    }
}

```



```

else if(buff_Is<503)Is=0;

if(Is<=0)
{
    Is=0;
    state_Is=0;
}
if(Is<=current_treshold)state_Is=1;
else if(Is>current_treshold)state_Is=2;

buff_Is=0;
buff_Vs=0;
datum1=0;

}
else if(buff_Vs<550 && buff_Vs>470)
{
    state_Vs=0;
    Vs=0;
    Is=0;
    buff_Is=0;
    buff_Vs=0;
    datum1=0;
}
else if(buff_Vs>825)
{
    Vs=0.277*buff_Vs+ 13.57;
    state_Vs=2;
    buff_Is=0;
    buff_Vs=0;
    datum1=0;
}
else if(buff_Vs<675)
{
    Vs=0.277*buff_Vs+ 13.57;
    state_Vs=4;
    buff_Is=0;
    buff_Vs=0;

```

```

    datum1=0;
}
}

if(datum2<=50)
{
    if(buff_It<=sensor_It)buff_It=sensor_It;
    if(buff_Vt<=sensor_Vt)buff_Vt=sensor_Vt;

    datum2++;
}
else //itung vt
{
    datum2=0;
    /*if(!digitalRead(S1))
    {
        lcd.setCursor(2,2);
        lcd.print(buff_Vr);
        lcd.setCursor(2,3);
        lcd.print(buff_Ir);
    }*/
    if (buff_Vt>675 && buff_Vt<825)
    {
        state_Vt=1;

        Vt=0.268*buff_Vt+ 6.57;

        if(buff_It>503 && buff_It<533)It=buff_It*0.0143 - 7.2097;
        else if(buff_It>=533 && buff_It<556)It=buff_It*0.0191 - 9.7665;
        else if(buff_It>=556 && buff_It<580)It=buff_It*0.0179 - 9.0917;
        else if(buff_It>=580)It=buff_It*0.0169 - 8.5154;
        else if(buff_It==502)It=0.00;
        else if(buff_It<502)It=0;

        if(It<=0)
        {
            It=0;
            state_It=0;
        }
    }
}

```

```

if(It<=current_treshold)state_It=1;
else if(It>current_treshold)state_It=2;

buff_It=0;
buff_Vt=0;
datum2=0;

}
else if(buff_Vt<550 && buff_Vt>470)
{
    state_Vt=0;
    Vt=0;
    It=0;
    buff_It=0;
    buff_Vt=0;
    datum2=0;
}
else if(buff_Vt>825)
{
    Vt=0.277*buff_Vt+ 13.57;
    state_Vt=2;
    buff_It=0;
    buff_Vt=0;
    datum2=0;
}
else if(buff_Vt<675)
{
    Vt=0.277*buff_Vt+ 13.57;
    state_Vt=4;
    buff_It=0;
    buff_Vt=0;
    datum2=0;
}

}

delay(5);
}

```

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN C

DATA SHEET

C.1 Data Sheet Sensor ACS712

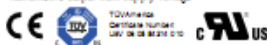


ACS712

*Fully Integrated, Hall Effect-Based Linear Current Sensor
with 2.1 kV_{RMS} Voltage Isolation and a Low-Resistance Current Conductor*

Features and Benefits

- Low-noise analog signal path
- Device bandwidth reset via the new FILTER pin
- 5 μ s output rise time in response to step input current
- 80 kHz bandwidth
- Total output error 1.5% at $T_A = 25^\circ\text{C}$
- Small footprint, low-profile SOIC package
- 1.2 m Ω internal conductor resistance
- 2.1 kV_{RMS} minimum isolation voltage from pins 1-4 to pins 5-8
- 5.0 V single supply operation
- 66 to 185 mV/A output sensitivity
- Output voltage proportional to AC or DC current
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis
- Ratio-metric output from supply voltage



Package: 8 Lead SOIC (suffix LC)



Approximate Scale 1:1

Description

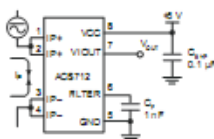
The Allegro[®] ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switched-mode power supplies, and overcurrent fault protection.

The device consists of a precise, low-offset, linear Hall sensor circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which is sensed by the integrated Hall IC and converted into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional voltage is provided by the low-offset, dropout-stabilized BiCMOS Hall IC, which is programmed for accuracy after packaging.

The output of the device has a positive slope ($>V_{OUTQ}$) when an increasing current flow through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4), which is the path used for current sensing. The internal resistance of this conductive path is 1.2 m Ω typical, providing low power

Continued on the next page...

Typical Application



Application 1: The ACS712 outputs an analog signal, V_{OUT} , that varies linearly with the uni- or bi-directional AC or DC primary sensed current, I_P , within the range specified. C_L is recommended for noise management, with values that depend on the application.

ACS712-05, Rev. 7

ACS712

Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kV RMS Voltage Isolation and a Low-Resistance Current Conductor

Description (continued)

box. The thickness of the copper conductor allows survival of the device at up to 5× overcurrent conditions. The terminals of the conductive path are electrically isolated from the sensor leads (pins 5 through 8). This allows the ACS712 current sensor to be used in applications requiring electrical isolation without the use of opto-isolators or other costly isolation techniques.

The ACS712 is provided in a small, surface-mount SOIC8 package. The leadframe is plated with 100% matte tin, which is compatible with standard lead (Pb) and unleaded circuit board assembly processes. Internally, the device is Pb-free, except for flip-chip high-temperature Pb-based solder balls, currently exempt from RoHS. The device is fully calibrated prior to shipment from the factory.

Selection Guide

Part Number	Packaging*	T _A (°C)	Optimized Range, I _S (A)	Sensitivity, Sens (V/A)
ACS712BLCTR-050-T	Tape and reel, 3,000 pieces/reel	-40 to 85	±5	185
ACS712BLCTR-20A-T	Tape and reel, 3,000 pieces/reel	-40 to 85	±20	100
ACS712BLCTR-30A-T	Tape and reel, 3,000 pieces/reel	-40 to 85	±30	85

*Contact Allegro for additional packaging options.

Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Supply Voltage	V _{DD}		5	V
Reverse Supply Voltage	V _{DD(Rev)}		-0.1	V
Output Voltage	V _{OUT}		5	V
Reverse Output Voltage	V _{OUT(Rev)}		-0.1	V
Reinforced Isolation Voltage	V _{ISO}	Pins 1-4 and 5-8 50 Hz, 1 minute, T _A = 25°C Voltage applied to leadframe (top pin), biased on IEC 60650	2100	V
Basic Isolation Voltage	V _{ISO(B)}	Pins 1-4 and 5-8 50 Hz, 1 minute, T _A = 25°C Voltage applied to leadframe (top pin), biased on IEC 60650	1500	V
Output Current Source	I _{OUT(sink)}		3	mA
Output Current Sink	I _{OUT(source)}		10	mA
Overcurrent Tolerance	I _{OL}	1 pulse, 100 ms	100	A
Nominal/Operating Ambient Temperature	T _A	Range B	-40 to 85	°C
Maximum Junction Temperature	T _{J(max)}		165	°C
Storage Temperature	T _{STG}		-65 to 170	°C

Parameter	Specification
FineLine Electric Shock	CAN/CSA-C22.2 No. 60950-1-03 UL 60950-1:2003 EN 60950-1:2001



Allegro MicroSystems, Inc.
1110 Northline Court
Folsom, CA 95630-8000
1-800-835-8900 www.allegromicro.com

Pinout diagram of the 8-pin DIP package:

- Pin 1: GND
- Pin 2: GND
- Pin 3: GND
- Pin 4: GND
- Pin 5: VCC
- Pin 6: VOUT
- Pin 7: FILTER
- Pin 8: GND

Number	Name	Description
1 and 2	IPH	Terminals for current being sensed; fused internally
3 and 4	IPL	Terminals for current being sensed; fused internally
5	GND	Signal ground terminal
6	RLTR	Terminals for external capacitor that sets bandwidth
7	VIQUT	Analog output signal
8	VSUP	Device supply voltage terminal

COMMON OPERATING CHARACTERISTICS¹ over full range of T_A , $C_R = 1 \text{ nF}$, and $V_{DD} = 5 \text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
ELECTRICAL CHARACTERISTICS						
Supply Voltage	V_{DD}		4.5	5.0	5.5	V
Supply Current	I_{DD}	$V_{DD} = 5.0 \text{ V}$, output open	—	10	13	mA
Output Capacitance Load	C_{LOAD}	VOUT to GND	—	—	10	nF
Output Resistive Load	R_{LOAD}	VOUT to GND	4.7	—	—	kΩ
Primary Conductor Resistance	$R_{PRIMARY}$	$T_A = 25^\circ\text{C}$	—	1.2	—	mΩ
Rise Time	t_r	$I_R = I_{R(max)}$, $T_A = 25^\circ\text{C}$, $C_{LOAD} = \text{open}$	—	5	—	μs
Frequency Bandwidth	f	−3 dB, $T_A = 25^\circ\text{C}$, I_R is 10A peak-to-peak	—	80	—	kHz
Nonlinearity	ϵ_{NL}	Over full range of I_R	—	1.5	—	%
Symmetry	ϵ_{SYM}	Over full range of I_R	95	100	102	%
Zero Current Output Voltage	$V_{OUT(0)}$	Bidirectional; $I_R = 0 \text{ A}$, $T_A = 25^\circ\text{C}$	—	$V_{DD} \times 0.5$	—	V
Power-On Time	t_{PO}	Output reaches 90% of steady-state level; $T_A = 25^\circ\text{C}$, 20 A present on leadframe	—	35	—	μs
Magnetic Coupling ²			—	12	—	G/A
Internal Filter Resistance ³	$R_{F(INT)}$		—	1.7	—	kΩ

¹ Device may be operated at higher primary current levels, I_R , and ambient T_A , and internal leadframe temperatures, T_L , provided that the Maximum Junction Temperature, $T_{J(max)}$, is not exceeded.

² $\pm 0.1 \text{ mT}$.

³ $R_{F(INT)}$ forms an RC circuit via the FILTER pin.

COMMON THERMAL CHARACTERISTICS¹

		Min.	Typ.	Max.	Units
Operating Internal Leadframe Temperature	T_L	−40	—	85	$^\circ\text{C}$
Junction-to-Lead Thermal Resistance ²	R_{JA}	Mounted on the AllegroA55K 712 evaluation board			
Junction-to-Ambient Thermal Resistance	R_{JA}	Mounted on the Allegro 55-0322 evaluation board, includes the power consumed by the board			
				5	$^\circ\text{C/W}$
				23	$^\circ\text{C/W}$

¹ Additional thermal information is available on the Allegro website.

² The Allegro evaluation board has 1500 mm² of 2 oz. copper on each side, connected to pins 1 and 2, and to pins 3 and 4, with thermal vias connecting the layers. Performance values include the power consumed by the PCB. Further details on the board are available from the Frequently Asked Questions document on our website. Further information about board design and thermal performance also can be found in the Applications Information section of this datasheet.



Allegro Microsystems, Inc.
1118 Northern Blvd.
Westborough, MA 01581-0001 U.S.A.
Tel: 508.853.8000 www.allegromicro.com

x05B PERFORMANCE CHARACTERISTICS $T_A = -40^{\circ}\text{C}$ to 85°C ; $C_D = 1\text{ nF}$, and $V_{DD} = 5\text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_{IN}		-5	-	5	A
Sensitivity	Sens	Over full range of I_{IN} , $T_A = 25^{\circ}\text{C}$	160	165	160	mV/A
Noise	$V_{NOISE(RMS)}$	Peak-to-peak, $T_A = 25^{\circ}\text{C}$, 165 mV/A programmed Sensitivity, $C_D = 47\text{ nF}$, $C_{OUT} = \text{open}$, 2 kHz bandwidth	-	21	-	mV
Zero Current Output Slope	ΔOutput	$T_A = -40^{\circ}\text{C}$ to 25°C	-	-0.28	-	mV/°C
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.08	-	mV/°C
Sensitivity Slope	ΔSens	$T_A = -40^{\circ}\text{C}$ to 25°C	-	0.054	-	mV/A/°C
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.008	-	mV/A/°C
Total Output Error ¹	E_{TOUT}	$I_{IN} = 5\text{ A}$, $T_A = 25^{\circ}\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_{IN} , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(max)}$, is not exceeded.

²Percentage of I_{IN} with $I_{IN} = 5\text{ A}$. Output filtered.

x20A PERFORMANCE CHARACTERISTICS $T_A = -40^{\circ}\text{C}$ to 85°C ; $C_D = 1\text{ nF}$, and $V_{DD} = 5\text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_{IN}		-20	-	20	A
Sensitivity	Sens	Over full range of I_{IN} , $T_A = 25^{\circ}\text{C}$	98	100	104	mV/A
Noise	$V_{NOISE(RMS)}$	Peak-to-peak, $T_A = 25^{\circ}\text{C}$, 100 mV/A programmed Sensitivity, $C_D = 47\text{ nF}$, $C_{OUT} = \text{open}$, 2 kHz bandwidth	-	11	-	mV
Zero Current Output Slope	ΔOutput	$T_A = -40^{\circ}\text{C}$ to 25°C	-	-0.34	-	mV/°C
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.07	-	mV/°C
Sensitivity Slope	ΔSens	$T_A = -40^{\circ}\text{C}$ to 25°C	-	0.017	-	mV/A/°C
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.004	-	mV/A/°C
Total Output Error ¹	E_{TOUT}	$I_{IN} = 20\text{ A}$, $T_A = 25^{\circ}\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_{IN} , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(max)}$, is not exceeded.

²Percentage of I_{IN} with $I_{IN} = 20\text{ A}$. Output filtered.

x30A PERFORMANCE CHARACTERISTICS $T_A = -40^{\circ}\text{C}$ to 85°C ; $C_D = 1\text{ nF}$, and $V_{DD} = 5\text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_{IN}		-30	-	30	A
Sensitivity	Sens	Over full range of I_{IN} , $T_A = 25^{\circ}\text{C}$	84	88	88	mV/A
Noise	$V_{NOISE(RMS)}$	Peak-to-peak, $T_A = 25^{\circ}\text{C}$, 88 mV/A programmed Sensitivity, $C_D = 47\text{ nF}$, $C_{OUT} = \text{open}$, 2 kHz bandwidth	-	7	-	mV
Zero Current Output Slope	ΔOutput	$T_A = -40^{\circ}\text{C}$ to 25°C	-	-0.38	-	mV/°C
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.08	-	mV/°C
Sensitivity Slope	ΔSens	$T_A = -40^{\circ}\text{C}$ to 25°C	-	0.007	-	mV/A/°C
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.002	-	mV/A/°C
Total Output Error ¹	E_{TOUT}	$I_{IN} = 30\text{ A}$, $T_A = 25^{\circ}\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_{IN} , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(max)}$, is not exceeded.

²Percentage of I_{IN} with $I_{IN} = 30\text{ A}$. Output filtered.

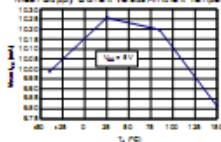


Allegro MicroSystems, Inc.
1110 Holloman Blvd.
Folsom, CA 95630-4000 U.S.A.
Tel: 916.351.7500 www.allegromicro.com

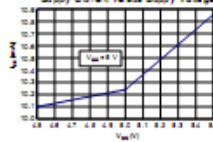
Characteristic Performance

 $I_s = 5A$, unless otherwise specified

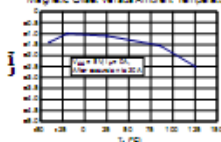
Mean Supply Current versus Ambient Temperature



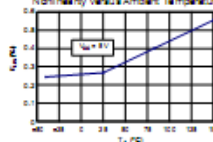
Supply Current versus Supply Voltage



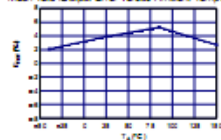
Magnetic Offset versus Ambient Temperature



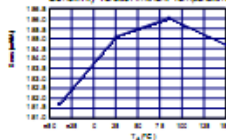
Nonlinearity versus Ambient Temperature



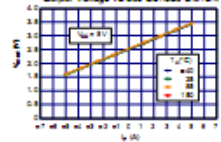
Mean Total Output Error versus Ambient Temperature



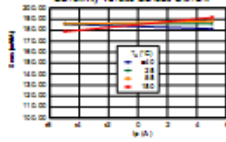
Sensitivity versus Ambient Temperature



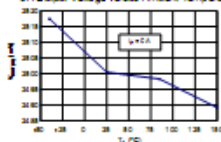
Output Voltage versus Sensed Current



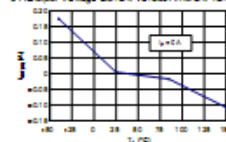
Sensitivity versus Sensed Current



0A Output Voltage versus Ambient Temperature



0A Output Voltage Current versus Ambient Temperature



Allegro MicroSystems, Inc.
118 North St. East
P.O. Box 910, Allentown, PA 18106-0910, U.S.A.
1-800-835-8949 www.allegromicro.com

C.2 Datasheet Arduino Mega 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical
Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms &
Conditions

Page 7

Environmental Policies
half sqm of green via Impatto Zero®

Page 7



radiospares

RADIONICS



Technical Specification

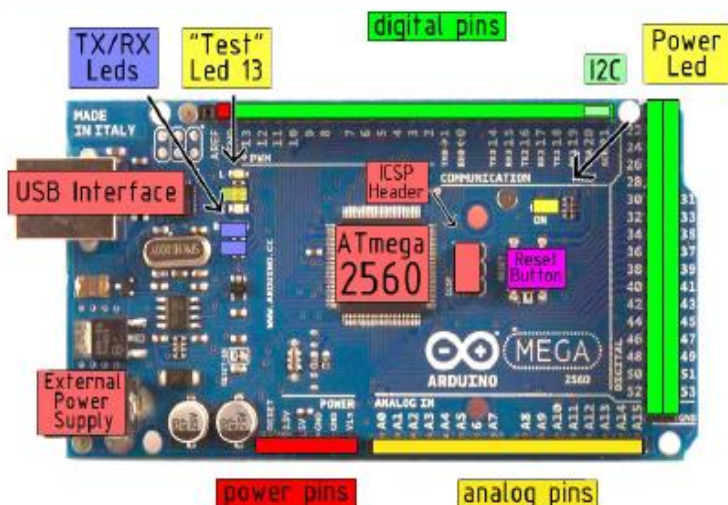


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 (Interrupt 0), 3 (Interrupt 1), 18 (Interrupt 5), 19 (Interrupt 4), 20 (Interrupt 3), and 21 (Interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 0 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C:** 20 (SDA) and 21 (SCL). Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



radiospares

RADIONICS



Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**



radiospares

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your skeeth you'll see something very close to the screenshot on the right.

In Tools>Board select MEGA

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



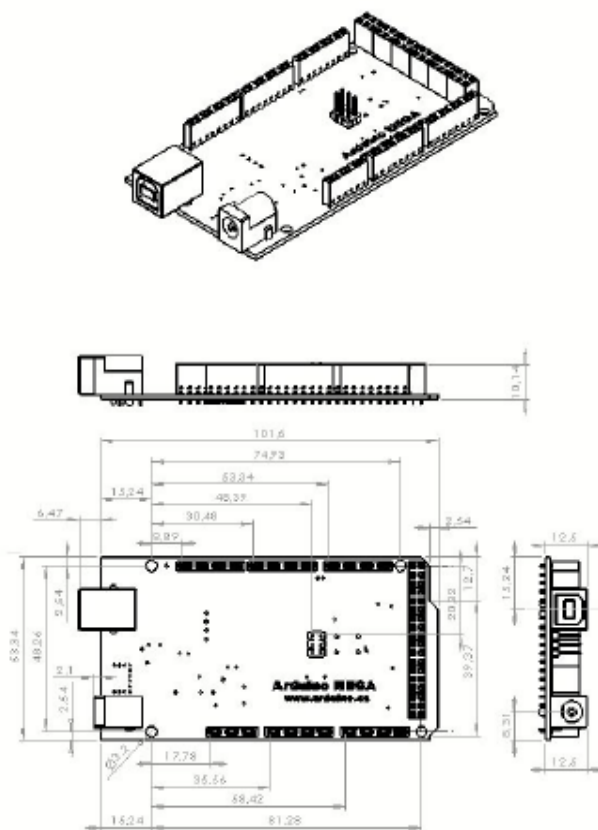
Blinking Led!



radiospares

RADIONICS





radiospares

RADIONICS



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mismanagement by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, the producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damage Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



radiospares

RADIONICS



RIWAYAT HIDUP PENULIS



Nama : Muhammad Ilham Fadlila
TTL : Surabaya, 18 Januari 1997
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Gunungsari Indah Blok
AB-49 Surabaya
Telp/HP : 081553977444
E-mail : *milhamfadlila@gmail.com*

RIWAYAT PENDIDIKAN:

1. 2003-2009 : SD Siti Aminah
2. 2009-2012 : SMP Negeri 3 Surabaya
3. 2012-2015 : SMA Negeri 6 Surabaya
4. 2015-2018 : Program Studi Elektro Industri,
Departemen Teknik Elektro Otomasi,
Fakultas Vokasi, Institut Teknologi
Sepuluh Nopember, Surabaya

PENGALAMAN KERJA

1. Kerja Praktek di PT. PLN (Persero) Area Pengatur Distribusi Jawa Timur, Ngagel, Surabaya
2. Kerja Praktek di Perum LPPNPI AirNav Indonesia, Juanda, Sidoarjo

PENGALAMAN ORGANISASI

1. Staff Departemen HMKI HIMAD3TEKTRO 2016-2017
2. Kabiro Relasi Kampus BEM FAKULTAS VOKASI 2018

-----Halaman ini sengaja dikosongkan-----